DUDLEY KNOX LIBRARY
NAVAL POSTGRADUATE SCHOOL
MONTEREY CA 93943-5101

Approved for public release; distribution is unlimited.

Theoretical Basis for State Vector Comparison, Relative Position Display, and Relative Position/Rendezvous Prediction

by

Lester Brown Makepeace III
Lieutenant, United States Navy
B.A., University of Maine, 1984

Submitted in partial fulfillment of the
requirements for the degrees of

MASTER OF SCIENCE IN APPLIED MATHEMATICS

and

MASTER OF SCIENCE IN ENGINEERING SCIENCE

from the

NAVAL POSTGRADUATE SCHOOL
December, 1993

REPORT DOCUMENTATION PAGE

Form Approved OMB No. 0704

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instruction, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188) Washington DC 20503.

1. AGENCY USE ONLY *(Leave blank)*	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	December 1993	Master's Thesis

4. TITLE AND SUBTITLE THEORETICAL BASIS FOR STATE VECTOR COMPARISON, RELATIVE POSITION DISPLAY, AND RELATIVE POSITION/RENDEZVOUS PREDICTION

5. FUNDING NUMBERS

6. AUTHOR(S) Lester B. Makepeace III

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Naval Postgraduate School
Monterey CA 93943-5000

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.

12a. DISTRIBUTION/AVAILABILITY STATEMENT

Approved for public release; distribution is unlimited.

12b. DISTRIBUTION CODE

A

13. ABSTRACT *(maximum 200 words)*

This thesis outlines the theoretical underpinnings used for the software designed to meet Detailed Technical Objectives 700-6 and 700-7 for the Space Shuttle Discovery mission STS-51. The primary goal was to compare state vector information produced by an on board GPS receiver and Discovery's computers, and provide real time display of the results. Because state vector information for the ORFEUS/SPAS payload was also available, relative position and rendezvous information between Discovery and ORFEUS/SPAS was made possible. Analysis of the various state vectors was used to produce a graphical display, in an operationally meaningful format, to the flight crew of Discovery.

14. SUBJECT TERMS

Fast Relative Position Display and Prediction, Terminal Rendezvous Algorithms.

15. NUMBER OF PAGES
94

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
Unclassified	Unclassified	Unclassified	UL

NSN 7540-01-280-5500

Standard Form 298 (Rev. 2-89)
Prescribed by ANSI Std. 239-18

ABSTRACT

This thesis outlines the theoretical underpinnings used for the software designed to meet Detailed Technical Objectives 700-6 and 700-7 for the Space Shuttle Discovery mission STS-51. The primary goal was to compare state vector information produced by an on board GPS receiver and Discovery's computers, and provide real time display of the results. Because state vector information for the ORFEUS/SPAS payload was also available, relative position and rendezvous information between Discovery and ORFEUS/SPAS was made possible. Analysis of the various state vectors was used to produce a graphical display, in an operationally meaningful format, to the flight crew of Discovery.

10313
H277547
c.1

TABLE OF CONTENTS

I. INTRODUCTION 1

II. STATE VECTOR COMPARISON 6

A. ORBITER STATE VECTOR GENERATION 6

B. TANS STATE VECTOR GENERATION 7

C. SAWTOOTH PLOTS 8

III. RELATIVE POSITTON DISPLAY (ORBITAL SYSTEM) . . . 11

A. VBAR/HBAR/RBAR COORDINATE SYSTEM 11

1. Definition 11

2. Derivation 12

a. Algorithm 13

B. SIGNIFICANCE OF VBAR/HBAR/RBAR 17

IV. RELATIVE POSITION DISPLAY (ORBITER FIXED SYSTEM) . 20

A. SHUTTLE FIXED COORDINATES 20

B. PITCH/YAW ANGLES 21

1. Definition 21

2. Derivation 23

C. FAST PITCH/YAW 25

V. RELATIVE MOTION PREDICTION 27

DUDLEY KNOX LIBRARY
NAVAL POSTGRADUATE SCHOOL
MONTEREY CA 93943-5101

A. PROPAGATION 27

B. f AND g FUNCTIONS 30

C. FUTURE PLOTS 34

D. FUTURE THRUST 35

E. RENDEZVOUS PREDICTION 37

VI. RENDEZVOUS SOLUTION 39

A. LINEARIZED EQUATIONS OF RELATIVE MOTION 39

1. Coordinate System 39

2. Equations of Motion 40

B. ALGORITHM 43

1. Rendezvous Initiation Thrust 43

2. Rendezvous Termination Thrust 47

C. APPLICATION 48

VII. RENDEZVOUS SOLUTION (REVISITED) 50

A. THE GAUSS PROBLEM 51

1. Formulation 51

2. Solution 52

3. p-Iteration Method 54

B. ALGORITHM 56

C. ADVANTAGES 58

VIII. POST FLIGHT DATA 61

A. ON ORBIT 61

1. State Vector Quality 61

2. Terminal Initiation 62
B. DEBRIEFING 65
C. GPS ACCURACY 67
D. TDRSS VISIBILITY 71

IX. CONCLUSIONS 73
A. FLIGHT CREW REMARKS 73
B. FOLLOW ON RESEARCH 76
1. Lambert Targeting 76
2. Inclusion of a Lambert Algorithm 78
3. Time Constrained Rendezvous Accessibility . 78
4. Drag Accelerations 79
5. Vernier Effect 80
6. Target Attitude 81
C. SUMMARY 81

REFERENCES 83

INITIAL DISTRIBUTION LIST 84

ACKNOWLEDGEMENTS

This thesis represents the work of one member from a very well integrated team. The tremendous accomplishment of deriving, designing, encoding, testing, and employing such a complex piece of software within six months, would have been impossible without the tireless efforts of all team members. I would like to express my thanks to Lieutenant Lee Barker (Software Engineer), Lieutenant Carolyn Tyler (Computer Graphics Design), Lieutenant Steve Rehwald (Researcher), for their cooperation and patience.

Clearly, without the support of NASA, the efforts of the NPS team could not have amounted to anything. For the crew of STS-51, the Astronaut Office, and everyone at the Johnson Space Center, their cooperation was greatly appreciated and critical to our success.

Most importantly, I would like to offer my heart felt gratitude to Mr. Tom Silva of The Telemetry Workshop Inc. Without his patience in teaching the "C" programming language, and constant involvement in the design of the NPS software, the project simply could not have succeeded.

Finally, I must thank my wife Rose. Her patience, understanding, and unrelenting support, are the keys to all of my successes. Without her, nothing is possible.

I. INTRODUCTION

On 12 September 1993, the Space Shuttle Discovery launched for STS-51. One of the experiments performed during this flight was entitled DTO (Detailed Technical Objective) 700. DTO 700 was actually a compilation of experiments related to the use of portable computers for on orbit navigation aids. Portions of the software for DTO 700 were produced by a design team from the Naval Postgraduate School (NPS).

The primary responsibility of the NPS software was to perform rudimentary state vector comparison with information obtained from two separate sources. However, in the development stage of the project, it became apparent that much more than basic state vector comparison was possible. Additional state vector sources and orbiter attitude information became available, which provided the means of presenting meaningful *operational* information to the crew of STS-51. This thesis presents the theoretical basis for software developed by the NPS team which processed the various sources of state vector and attitude information into formats that were meaningful to Discovery's crew.

The secondary payload for STS-51 was the Shuttle Pallet Satellite (SPAS) carrying the ORFEUS payload. Since SPAS was designed to operate in proximity with Discovery, it produced a data stream that was continually data linked to Discovery

and available to the NPS software via Discovery's main computers. This data stream contained two separate sources of state vector information for SPAS. The first was produced by a GPS receiver, while the second was the output of orbit propagation software resident in computers located on the satellite. A second GPS receiver (a portable Trimble Navigation TANS GPS receiver) was on board Discovery providing orbiter state vector information. Discovery's own computers also produced state vectors for the orbiter and SPAS as well as orbiter attitude information. These sources of information provided the inputs for the derivations presented.

The primary responsibility of the NPS software was to compare the orbiter state vector information produced by the portable TANS GPS receiver, to that produced by Discovery's computers. The GPS information was ported directly to the GRID 1530 386/10 laptop computer being used for the comparison. The orbiter generated state vector information was read from information being downlinked to ground controllers. Tapping into this data stream provided the means of reading *SPAS* state vector information, as well as orbiter attitude information. The additional information provided by SPAS allowed the original scope of DTO 700 to be expanded to include displaying information with operational significance.

The executable program that flew aboard STS-51 contains three families of routines that provide information to the flight crew. The first group, called the sawtooth plots,

display magnitude differences between various parts of the input state vectors. This family of plots were designed to satisfy the primary responsibility of the NPS software.

The second family, called the RBAR/VBAR plots, are designed to show relative motion between spacecraft operating in proximity. This type of plot is used by the astronauts in planning their mission. The target spacecraft is placed in the center of a local vertical/circular coordinate system, while the pursuing spacecraft's position is displayed graphically by an altitude and downrange difference. The out of plane error is shown in an alphanumeric format to complete the three dimensional information. It is important to note that this is not a rectilinear coordinate system, however information displayed with this method provides a very intuitive feel of relative orbital motion.

The third family, called the Pitch/Yaw plots, was created as a means of providing information to the flight crew to assist in locating SPAS. This is accomplished by creating a vector to SPAS and then transforming it into a Shuttle based coordinate system. The information is finally displayed in terms of pitch and yaw angles.

Due to the proximity operations with SPAS, this thesis also addresses rendezvous solutions for spacecraft in similar orbits. In testing the software that was to fly, the need to maneuver the simulated Shuttle orbiter arose. Short routines used to apply velocity changes to the Shuttle's state vector

were created. These, however, were not designed to provide the velocity changes to apply for a given maneuver. The linearized relative equations of motion as presented in Reference 1 were solved in order to determine the velocity changes needed to initiate and terminate a rendezvous. The solutions to these equations are derivatives with respect to a non-inertial frame, requiring great care in transforming to the inertial frame. Only the initial velocity change solution has been incorporated in the NPS software, and this was disabled in the *primary* executable that flew aboard STS-51.

The rendezvous information that *was* incorporated in the executable that flew aboard STS-51 did not in itself produce a solution for rendezvous, but rather produced a prediction of relative position based on user input velocity changes in the Shuttle's coordinate system. The prediction was based on classical elements and the "f" and "g" functions, which do not account for accelerations other than those associated with the classic two body problem. However, for similar orbits, perturbing accelerations have similar effects and thus have minimal effect on relative motion.

The mathematics and physics of this thesis are not difficult to follow. The significance of this work lies in the applicability of commonly understood principles for a very relevant purpose. Flying an aircraft is very intuitive, however, flying a spacecraft requires knowledge of orbital mechanics and dynamics.

The software that flew aboard STS-51 automated many of the principles of these disciplines, and gave the crew of STS-51 a graphical presentation of their current situation, as well as their history, and a means of predicting future motion.

II. STATE VECTOR COMPARISON

The primary purpose of DTO 700 was to provide on orbit state vector comparison between orbiter generated state vectors and state vectors generated by the TANS portable GPS receiver. A state vector is composed of two cartesian vectors and a time element. The vectors represent the position and velocity of the prescribed spacecraft and are expressed in an inertial coordinate system known as M50. Throughout this treatment, the assumption is that state vectors being compared have the same time stamp. In reality, this rarely occurs. To account for unmatched times of state vectors, a Cowell integrator is used to propagate one of the state vectors until the state vectors are concurrent.

A. ORBITER STATE VECTOR GENERATION

The motivation for the state vector comparison is the belief that orbiter derived state vectors can accrue errors relatively quickly. The orbiter produces state vectors with on board computers by using orbit propagation software known as the "Super G". Periodically, ground controllers uplink an updated state vector to the orbiter derived from information collected by ground tracking stations. This state vector serves as the new initial condition for the Super G propagator, which provides a continuous stream data based on

the latest initial conditions. As with any numerical propagator, error is expected to accrue. Unfortunately, due to computational hardware limitations, the Super G is essentially a low fidelity propagator, which results in the possibility of increasingly large state vector errors during periods between state vector updates.

B. TANS STATE VECTOR GENERATION

The TANS portable GPS receiver generated the state vectors against which the Super G derived state vectors were compared. A GPS receiver can produce a state vector based on the information received from any four GPS satellites at a given instant. Given a period of favorable geometry with respect to GPS satellites, the TANS receiver produces a continuous stream of state vectors which have a bounded error. Initial data [Ref. 2] indicates accuracies within 100 meters in position and on the order of meters per second in velocity.

In addressing the requirements of DTO 700, the first responsibility of software created by the Naval Postgraduate School was to use these GPS derived state vectors, characterized by bounded error, to demonstrate the rate at which state vectors produced by the Super G degrade, and begin to validate the use of GPS as an on orbit navigation aid.

C. SAWTOOTH PLOTS

These plots derived their name from the expected form of their output data. Assuming the uplinked orbiter state vector's accuracy, it was expected that as the Super G propagated away from 'truth', the difference of the orbiter and TANS state vectors should become more pronounced, while at the instant a new state vector was uplinked, the difference should be minimized. To display this, a very simple algorithm comparing the difference vectors, in position and velocity, is presented.

Let $\vec{r}_T$ and $\vec{v}_T$ represent the position and velocity vectors produced by the TANS receiver, while $\vec{r}_o$ and $\vec{v}_o$ represent the position and velocity vectors produced by the orbiter. The difference vectors ($\vec{r}_d$ and $\vec{v}_d$) are given by

$$\begin{aligned} \vec{r}_d &= \vec{r}_o - \vec{r}_T \\ \vec{v}_d &= \vec{v}_o - \vec{v}_T \end{aligned} \qquad (2.1)$$

The magnitudes of these difference vectors, given by

$$\begin{aligned} r_d &= \|\vec{r}_d\| \\ v_d &= \|\vec{v}_d\| \end{aligned} \qquad (2.2)$$

are plotted against time to show the relative behavior of the two sources.

Figures 2.1 and 2.2 show the display screens of r_d and v_d plots that were produced in simulation. For this simulation, a corrected uplinked state vector is created by matching the orbiter state vector with the simulated TANS state vector. The

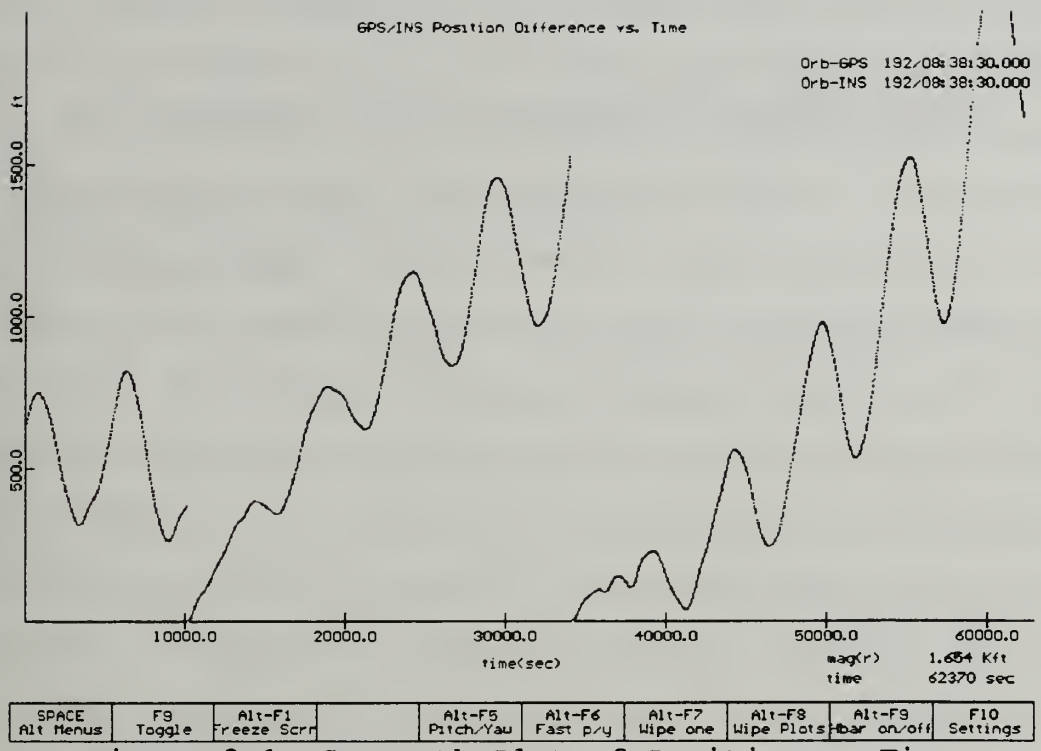

Figure 2.1 Sawtooth Plot of Position vs Time

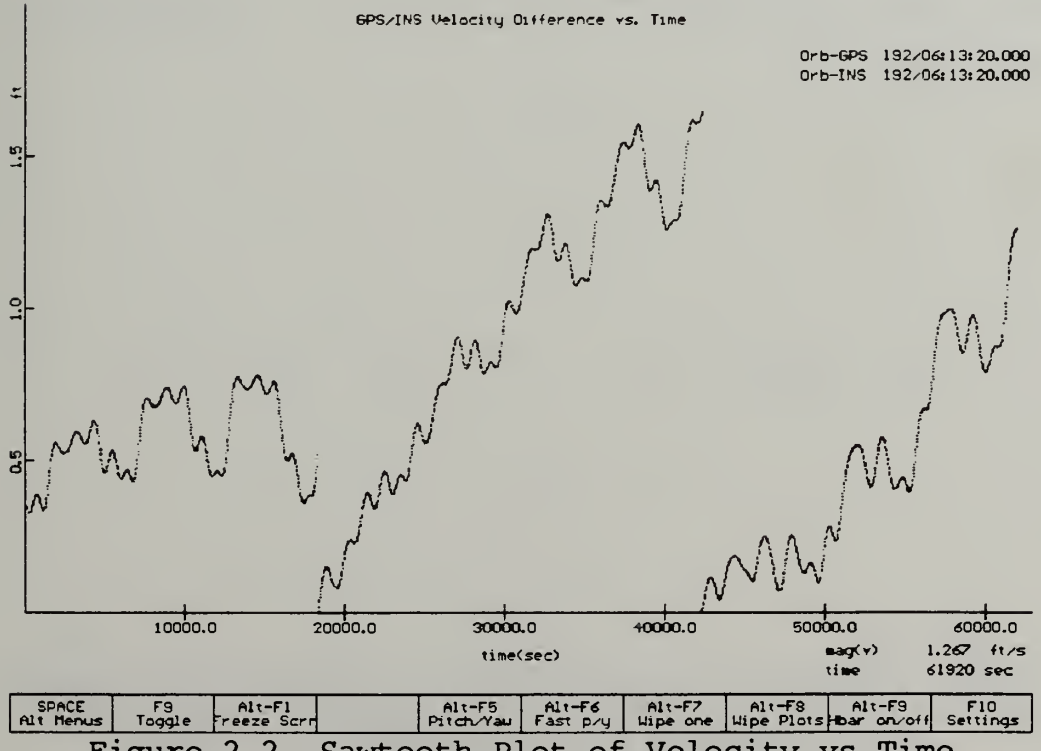

Figure 2.2 Sawtooth Plot of Velocity vs Time

drift between state vectors is achieved by using a lower fidelity propagator for the orbiter state vector than for the TANS state vector. Achieving a perfect match between state vectors at the moment of uplink is not actually expected. It may be noted that times corresponding to the low point of the sawteeth in Figures 2.1 and 2.2 are different. This is because the plots were produced with separate simulations, and is not due to a miscorrelation between position and velocity updates.

Although the original intent was to merely compare TANS and orbiter generated state vectors, the software produced by the NPS team that flew aboard STS-51 provided the means to input any pair of state vectors for this comparison. Specifically, a similar pair of propagated and GPS derived state vectors for SPAS were also available.

III. RELATIVE POSITION DISPLAY (ORBITAL SYSTEM)

A. VBAR/HBAR/RBAR COORDINATE SYSTEM

1. Definition

The VBAR/HBAR/RBAR (also referred to as RBAR/VBAR) coordinate system is a local vertical/circular (LVC) frame used for displaying the relative position of two orbiting bodies. This system is precisely the one used by NASA planners in planning shuttle maneuvers in close proximity with a target satellite. Figure 3.1 shows the background screen used for displaying RBAR/VBAR position. It is critically important to

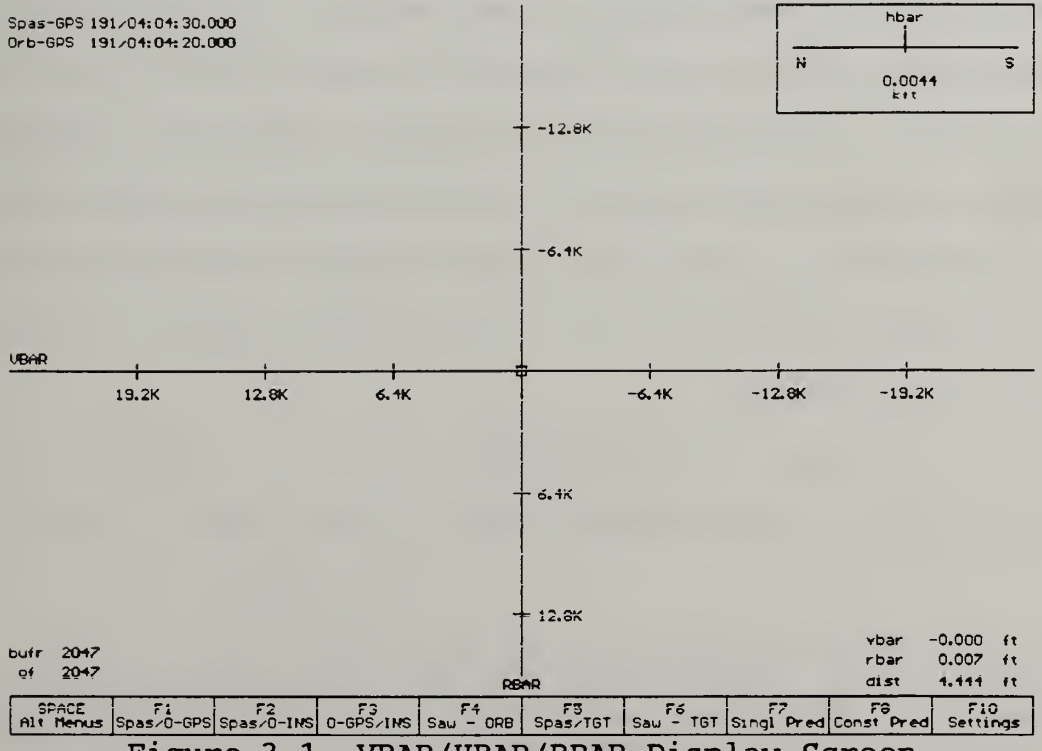

Figure 3.1 VBAR/HBAR/RBAR Display Screen

note that this is not a rectilinear system, and thus no convenient coordinate transformation matrix can be derived.

The center of the coordinate system is the target spacecraft (SPAS for STS-51). The relative position of the chaser spacecraft (Space shuttle Discovery for STS-51) is then plotted relative to the target. The horizontal axis is called the VBAR. The name is derived from the fact that for circular (or near circular) orbits, this axis is generally aligned with the target's velocity vector. The vertical axis is called the RBAR. So named because it is defined by the target's position vector. Displacement along the RBAR is measured positively toward the earth and represents an altitude difference between the target and chaser. Displacement along the VBAR is measured positively in the direction of travel of the target and represents a curvilinear distance ahead or behind the target measured at the target's altitude. HBAR is displayed alphanumerically in the upper right hand corner of the screen. It represents a north/south[1] distance from the orbital plane of the target measured in kft for shuttle orbits.

2. Derivation

Although the RBAR/VBAR coordinate system is not rectilinear, it closely parallels a system that is, which is

---

[1]Given the orbit for STS-51 had an inclination of approximately 28°, HBAR displacement is actually not purely in a north/south direction, however north/south is used to denote the general direction of displacement.

often referred to as local vertical/local horizontal (LVLH). There are alternate conventions for determining the direction of positive axes in LVLH, but for consistency, these directions will parallel those of the RBAR/VBAR frame.

Coincident with the construction of the RBAR/VBAR frame, the corresponding LVLH frame is also derived. Since LVLH is rectilinear, it can be specified by a transformation matrix from the inertial system in which the input state vectors are displayed. This matrix, often referred to as a direction cosine matrix (DCM), will be denoted by the symbol ${}^{L}C^{I}$. The symbol is read "the transformation matrix from I (inertial) to L (LVLH)". The "I" appears on the right hand side of the symbol because a column vector in inertial coordinates must be placed on the right side of this matrix for matrix multiplication to produce the corresponding vector in LVLH coordinates. Since the source code for the flight software is written in the "C" programming language, where zero subscripting is used, the elements of this matrix are written

$$ {}^{L}C^{I} = \begin{bmatrix} C_{00} & C_{01} & C_{02} \\ C_{10} & C_{11} & C_{12} \\ C_{20} & C_{21} & C_{22} \end{bmatrix} \qquad (3.1) $$

a. Algorithm

In producing the RBAR/VBAR coordinates for display, it is necessary to construct ${}^{L}C^{I}$ from input state

vector data. State vectors for the target and chaser are provided which contain the respective position and velocity vectors displayed in inertial space. Given the position and velocity vectors of the target ($\vec{\mathbf{r}}_t$, $\vec{\mathbf{v}}_t$), and the position and velocity vectors of the chaser ($\vec{\mathbf{r}}_c$, $\vec{\mathbf{v}}_c$), the RBAR coordinate of the chaser is simply given by the difference in the magnitudes of the position vectors

$$RBAR = \|\vec{r}_t\| - \|\vec{r}_c\| \tag{3.2}$$

The target position vector defines the "z" axis of the LVLH frame. Recalling that positive displacement is toward the earth, a unit vector along the "z" axis is created by negating the normalized target position vector

$$\hat{z} = -\frac{\vec{r}_t}{\|\vec{r}_t\|} \tag{3.3}$$

This unit vector corresponds to the last column of ${}^{L}C^{I}$.

The middle column of ${}^{L}C^{I}$ is generated by the angular momentum vector of the target's orbit. This vector is given by the cross product of the target's position and velocity vectors

$$\vec{h}_t = \vec{r}_t \times \vec{v}_t \tag{3.4}$$

A unit vector perpendicular to the orbit plane and along the "y" axis can now be created by normalizing this vector. However, keeping in mind the desire to have the "x" axis point

ahead of the spacecraft, the negated normalized angular momentum is used

$$\hat{y} = -\hat{h}_t = -\frac{\vec{H}_t}{\|\vec{H}_t\|} \tag{3.5}$$

corresponding to the middle column of ${}^{L}C^{I}$.

Recall that the HBAR component represents the out of plane component of $\vec{r}_c$. As the orbital plane contains the origin of the inertial system in which $\vec{r}_c$ is measured, HBAR is given by the projection of $\vec{r}_c$ onto $\hat{h}_t$[2]

$$HBAR = \hat{h}_t \cdot \vec{r}_c \tag{3.6}$$

The left column of ${}^{L}C^{I}$ is produced from the orthonormal vectors $\hat{y}$ and $\hat{z}$ by

$$\hat{x} = \hat{y} \times \hat{z} \tag{3.7}$$

completing the matrix ${}^{L}C^{I}$

$${}^{L}C^{I} = [\hat{x}\ \hat{y}\ \hat{z}] \tag{3.8}$$

The VBAR component represents an angular displacement ahead or behind the target spacecraft. To determine this value, the projection of $\vec{r}_c$ onto the target's orbital plane must be found. Given HBAR is the magnitude of the out of plane displacement and $\hat{h}_t$ is a vector in the out of

---

[2]This choice for HBAR appears inconsistent with a right handed coordinate system. Since the RBAR/VBAR system is not rectilinear, this is of little consequence. Some references, however, may choose positive HBAR opposite the angular momentum vector.

plane direction, the in plane component of $\vec{r}_c$ ($\vec{r}_{cop}$) is given by

$$\vec{r}_{cop} = \vec{r}_c - HBAR\hat{h} \tag{3.9}$$

The angular displacement between $\vec{r}_{cop}$ and $\vec{r}_t$ is then

$$\eta = \arccos\left(\frac{\vec{r}_{cop} \cdot \vec{r}_t}{\|\vec{r}_{cop}\| \|\vec{r}_t\|}\right) \tag{3.10}$$

Unfortunately, computational machines will produce a positive number for Equation 3.10 due to the proximity of the spacecraft, resulting in an angular displacement without the corresponding direction required to determine the sign of VBAR. This problem is the motivation for creating ${}^{L}C^{I}$ while computing the RBAR/VBAR coordinates. Consider the vector from the target to the chaser

$$\vec{r}_{dif} = \vec{r}_c - \vec{r}_t \tag{3.11}$$

Transforming this vector into LVLH coordinates via

$$\left[\vec{r}_{dif}\right]_L = {}^{L}C^{I}\left[\vec{r}_{dif}\right]_I \tag{3.12}$$

produces a vector whose "x" coordinate must have the same sign as that of VBAR. Modifying the sign of η to match the sign of the "x" coordinate given by Equation 3.12, VBAR is then

$$VBAR = \eta \cdot \|\vec{r}_t\| \tag{3.13}$$

which is an arc length corresponding to the in plane angular displacement relative to the target.

B. SIGNIFICANCE OF VBAR/HBAR/RBAR

As previously stated, this is the coordinate system used by NASA for planning proximity operations. The reason for this is that, to first order, the angular momentum of an orbit is constant. For near circular orbits, position and velocity vectors are nearly perpendicular, and therefor from Equation 3.4, an increase in one must correspond to a decrease in the other. When two spacecraft fly in close proximity, their orbits must have nearly equal angular momentum vectors. In this case, a displacement above the VBAR corresponds to a larger chaser position vector than target position vector, leading to a smaller velocity, and a resulting backward drift of the chaser relative to the target.

Figure 3.2 shows the effect of a posigrade burn for the chaser at an instant when the target and chaser have identical state vectors. Intuitively, the chaser is expected to move ahead of the target. However, this is true only for an instant. The increase in velocity will correspond to an increase in angular momentum, which in turn corresponds to an increase in the semi-major axis of the orbit. The result is to cause the chaser to drift above the VBAR, and therefor begin to fall behind the target. The bouncing phenomenon shown in Figure 3.2 is due to the fact that the point where the velocity increase occurs is coincident with both orbits. The target, with the slightly smaller semi-major axis, has a slightly smaller orbital period, thus reaching this point

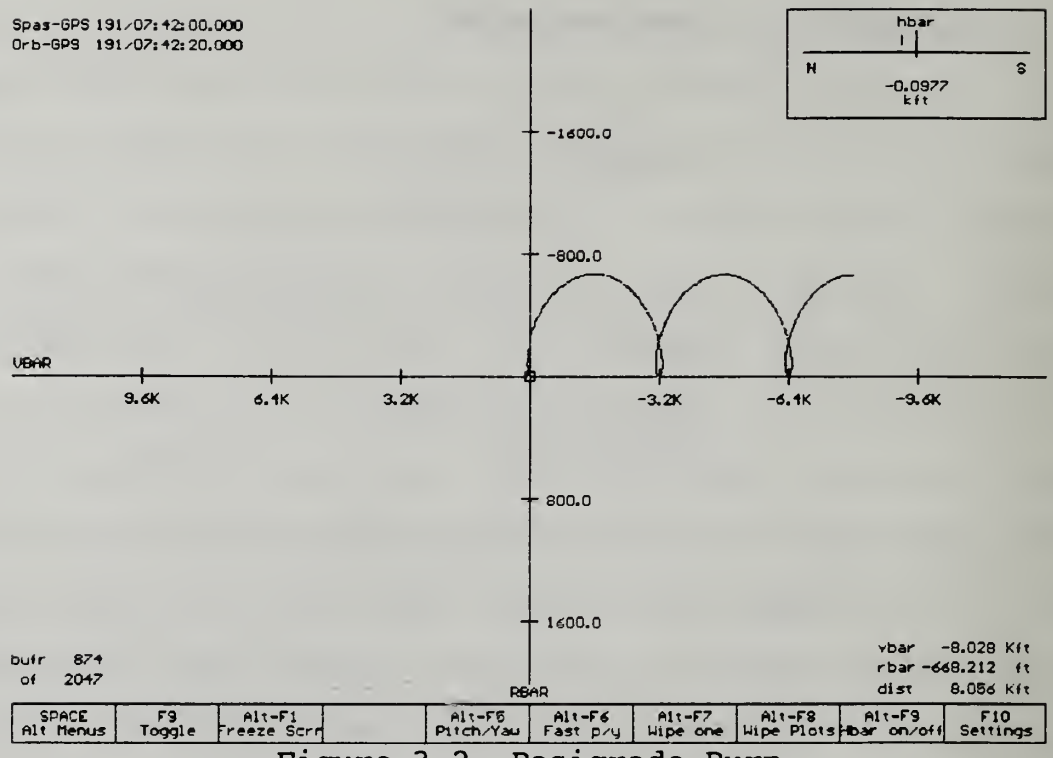

Figure 3.2 Posigrade Burn

sooner than does the chaser. The chaser must, however, continue to pass through this point every orbit. If the target orbit is considered to be purely circular, the additional velocity given to the chaser has effectively caused it to have a slightly eccentric orbit with a radius of perigee equal to the circular radius of the target.

Figure 3.3 shows a similar situation for a thrust in the opposite direction. The argument is the same as for a posigrade burn, however in this case the angular momentum is decreased, producing a decrease of the chaser's semi-major axis and period. The effect is to cause the spacecraft to "bounce" forward under the VBAR.

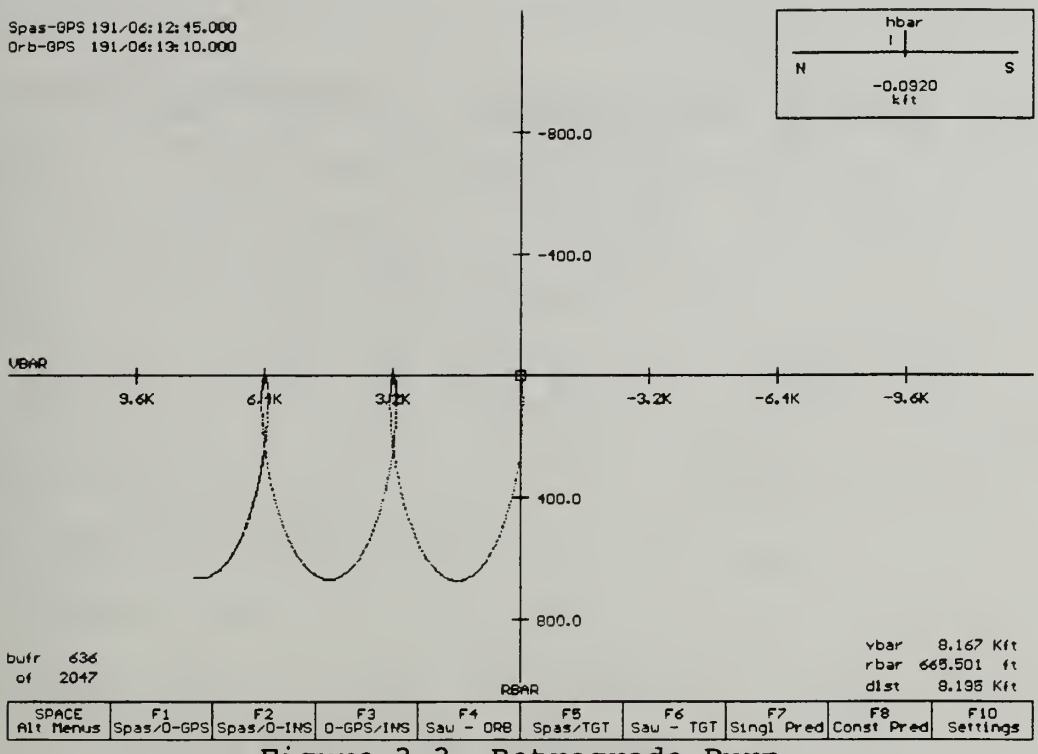

Figure 3.3 Retrograde Burn

Proximity operations for the Shuttle are all designed with displacements above and below the VBAR to cause the desired drift. Each "bounce" represents one orbit, which provides an inherent time reference.

Alphanumeric display of HBAR is useful for the following scenario. Ideally, the commander of a mission would prefer to match the orbital plane of a given target exactly. This requires an occasional thrust to void any out of plane motion. If the orbital planes are not matched, HBAR will cycle back and forth across the zero position. At an instant when HBAR hits zero, the orbiter is in the orbital plane of the target, representing an optimal time when a thrust should be applied to remove the out of plane motion.

IV. RELATIVE POSITION DISPLAY (ORBITER FIXED SYSTEM)

It is often convenient to express positions in a reference frame fixed on the orbiter. This gives the crew the most intuitive feel for the information presented.

At the request of the crew of STS-51, a means of presenting the position of SPAS relative to the orbiter was produced. The request for this type of display was motivated by the size of the crew. Having only a five man crew, all crew members slept at the same time. The crew requested the ability to locate SPAS when they awoke, so they might know where to point other sensors to acquire the target.

A. SHUTTLE FIXED COORDINATES

The coordinate system fixed to the orbiter is aligned such that, if the orbiter were flying like an airplane, it would be aligned with the LVLH system derived in the previous chapter, though this is rarely the case. The positive "x" axis points directly out of the nose of the orbiter, positive "z" points out of the belly of the orbiter, thus forcing positive "y" to point out of the right wing. Because this is a rectilinear system, a transformation matrix relating this system to the inertial coordinate system exists.

Part of the downlink data stream provided to the software was a quaternion known as QBI (Quaternion Body/Inertial). A

quaternion is a four position vector containing information relating two coordinate systems. Quaternions are used not only for their compactness, but also because of their convenience when used in attitude control algorithms. The quaternion QBI relates the inertial coordinate system M50 to the orbiter fixed or "body" coordinate system. Since we will not be using QBI for attitude control algorithms, we have no need to perform quaternion algebra, and will immediately create the necessary transformation matrix from QBI.

Given

$$\vec{QBI} = \begin{bmatrix} q_1 \\ q_2 \\ q_3 \\ q_4 \end{bmatrix} \tag{4.1}$$

the matrix transformation from inertial to body coordinates (referred to as ${}^{B}C^{I}$) is given by [Ref. 3]

$${}^{B}C^{I} = \begin{bmatrix} q_1^2 + q_2^2 - q_3^2 - q_4^2 & 2(q_2 q_3 - q_1 q_4) & 2(q_2 q_4 + q_1 q_3) \\ 2(q_2 q_3 + q_1 q_4) & q_1^2 - q_2^2 + q_3^2 - q_4^2 & 2(q_3 q_4 - q_1 q_2) \\ 2(q_2 q_4 - q_1 q_3) & 2(q_3 q_4 + q_1 q_2) & q_1^2 - q_2^2 - q_3^2 + q_4^2 \end{bmatrix} \tag{4.2}$$

B. PITCH/YAW ANGLES

1. Definition

Given ${}^{B}C^{I}$, any vector expressed in inertial coordinates can be transformed into body coordinates. Specifically, the vector from the orbiter to the target can be expressed in body

coordinates. However, a vector is still only three real numbers, and thus does not offer much intuitive feel for the position. To provide intuition, this vector is translated into rotation angles through which to put the orbiter so as to point the nose of the orbiter on the target. This does not imply that the crew should maneuver the shuttle to see the target, but rather perform the maneuvers mentally to imagine where the orbiter would then be pointing.

Figure 4.1 shows a typical display screen for the Pitch/Yaw plots. The sequence shown in Figure 4.1 suggests

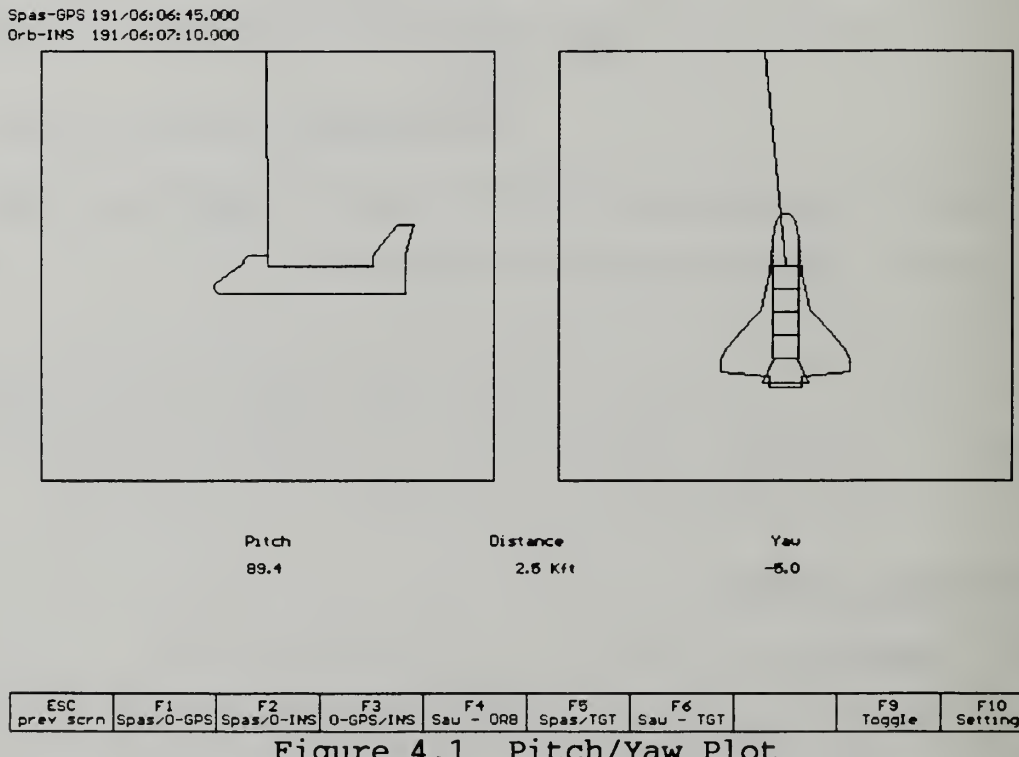

Figure 4.1 Pitch/Yaw Plot

first pitching the orbiter 89.4°; then from the new position, yaw the orbiter -5.0°. The final position achieved points the

nose of the orbiter directly at the target. This may not seem any more intuitive than a vector to someone unfamiliar with aviation. However, to an aviator, this clearly implies the target is almost directly overhead, slightly to the left.

The choice of performing a pitch maneuver, then a yaw maneuver, is not arbitrary. Most orbiter maneuvers near a target spacecraft are performed with the payload bay (top of the orbiter) pointing toward the target. If the pitch is exactly 90°, and a yaw maneuver had been chosen first, any value for yaw would have been acceptable. Thus pitching first is chosen with the knowledge that pitch is near 90°, eliminating the singularity present by trying to yaw first. Had the expected position of the target been near the orbiters "x"-"y" plane, an initial yaw maneuver would be more appropriate.

2. Derivation

The first step in determining a position relative to the orbiter is to produce a vector from the chaser (orbiter) to the target ($\vec{r}_{ct}$) via

$$\vec{r}_{ct} = \vec{r}_t - \vec{r}_c \qquad \textbf{(4.3)}$$

Recall that the position vectors for the target and chaser are expressed in inertial coordinates, and therefor Equation 4.3 gives $\vec{r}_{ct}$ in inertial coordinates. To express this vector in body coordinates, simply apply the coordinate transformation matrix ${}^{B}C^{I}$ created from the quaternion QBI.

$$\left[\vec{r}_{ct}\right]_B = {}^{B}C^{I}\left[\vec{r}_{ct}\right]_I \qquad (4.4)$$

Once $\vec{r}_{ct}$ is expressed in body coordinates, pitch and yaw angles can be produced. Figure 4.2 shows the geometry of problem. The parallelogram anchored at the origin has sides x, y, and z which are the coordinates of $\vec{r}_{ct}$. The vector $\vec{r}_p$ is the projection of $\vec{r}_{ct}$ onto the "x"-"z" plane and is the intermediate orientation to be achieved after the pitch maneuver. Let Φ be the angle between $\vec{r}_p$ and the "x"-"y" plane. Then Φ is the pitch angle sought and is given by the relationship

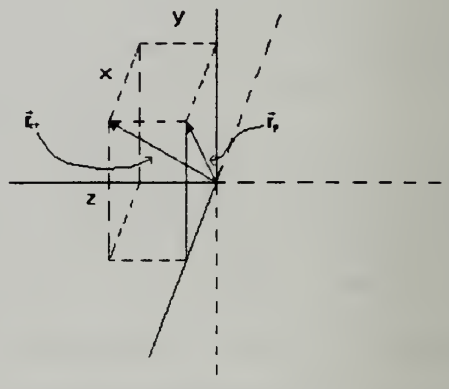

Figure 4.2 Pitch/Yaw Geometry

$$\tan\Phi = \frac{z}{x} \qquad (4.5)$$

Let Θ be the angle between $\vec{r}_{ct}$ and $\vec{r}_p$. Then Θ represents the yaw angle required to complete the maneuver. Since we know the magnitude of $\vec{r}_p$ is given by

$$r_p = \|\vec{r}_p\| = \sqrt{x^2 + z^2} \qquad (4.6)$$

then Θ is given by the relationship

$$\tan\Theta = \frac{Y}{r_F} \qquad (4.7)$$

As with the standard spherical coordinate system from analytic geometry, defining two angles establishes a direction toward a point, but not a magnitude. Thus the magnitude of the vector $\vec{r}_{ct}$ is also presented on the Pitch/Yaw Plot display screen.

C. FAST PITCH/YAW

The Pitch/Yaw algorithm was provided for STS-51 with graphics as shown in Figure 4.1, and also alphanumerically on the RBAR/VBAR screen upon request of the user. Figure 4.3 shows this display in the upper left corner.

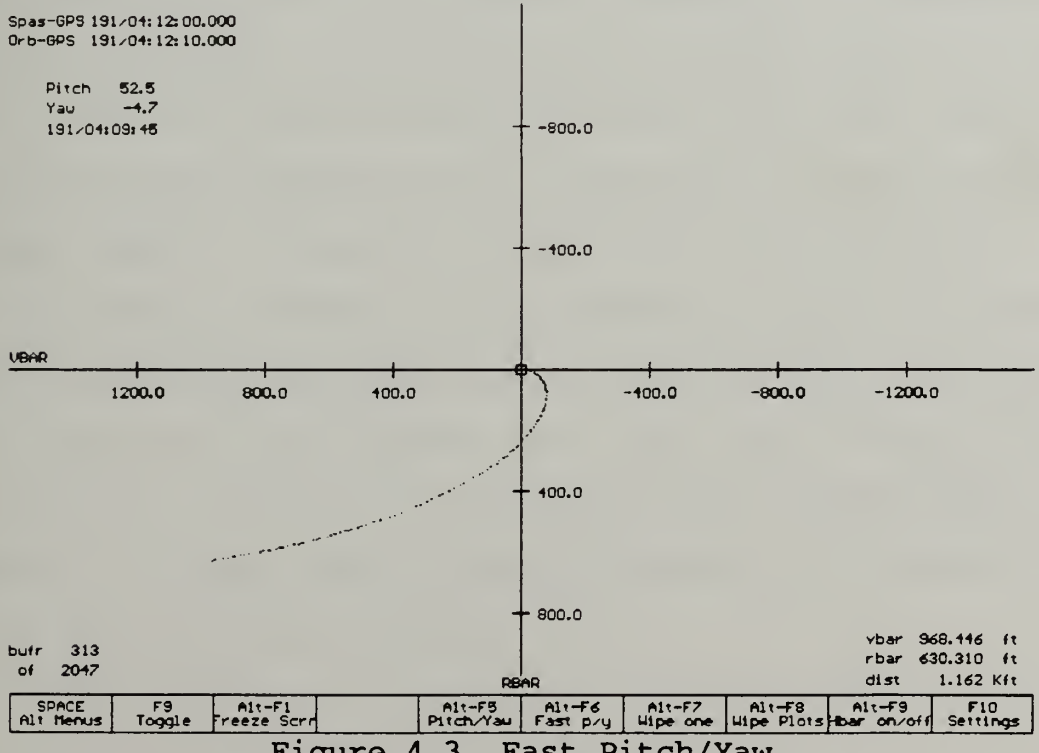

Figure 4.3 Fast Pitch/Yaw

Clearly, the RBAR/VBAR plots offer the crew the most situational awareness of the three families of plots presented thus far. However, while observing the RBAR/VBAR screen, the crew expressed a desire to have access to the output from the Pitch/Yaw algorithm. The Fast Pitch/Yaw option was created to provide situational awareness in the orbital reference frame as well as the shuttle fixed reference frame.

V. RELATIVE MOTION PREDICTION

A. PROPAGATION

Input for each algorithm presented are two state vectors which define two orbits. To predict the relative motion of the two bodies, the Cowell propagator mentioned earlier could be invoked to produce future state vectors, which in turn could be used in the relative position algorithms discussed in Chapter III. Unfortunately, as with any high fidelity propagator, the Cowell propagator is computationally intensive. To produce 40 predicted points would take 80 calls to the Cowell routine. This would not pose a problem if performed on a very fast computer. However, the computer in which these algorithms reside has a 10 MHz, 386 processor. This relatively slow machine does not offer the luxury of invoking a high fidelity Cowell propagator 80 times for every screen update, and therefor another solution was sought.

One obvious solution to this problem, is to simply reduce the fidelity of the Cowell routine. Flags could be set within the algorithm to account for only the central body acceleration, neglecting accelerations due to higher spherical harmonics or drag. This raises the question of accuracy for the predicted relative motion.

At typical shuttle altitudes, the major perturbing acceleration for an orbit is due to the second zonal harmonic, often referred to as the J_2 term. The acceleration due to J_2 is roughly two orders of magnitude larger than any other perturbing acceleration. Effects due to J_2 on an orbit viewed in inertial space are often evident within one orbit, thus neglecting this term may initially seem unwise. There is, however, an interesting property of all spherical harmonics that justifies their exclusion for the purposes of predicting relative motion.

Accelerations due to spherical harmonics are a function of position relative to the center of the pseudo-spherical body about which the satellite is orbiting. The potential of the earth's gravity field, or geopotential, is given by

$$V_\oplus = -\frac{\mu_\oplus}{r}\left[1 + \sum_{n=2}^{N}\sum_{m=0}^{n}\left(\frac{a}{r}\right)^n \bar{P}_{nm}(\sin\phi)\left(\bar{C}_{nm}\cos m\lambda + \bar{S}_{nm}\sin m\lambda\right)\right] \quad (5.1)$$

where the parameters are defined to be:

$V_\oplus$ = Geopotential function

$\mu_\oplus$ = Gravitational constant of the earth

r = Magnitude of the radius vector

a = Semi-major axis of the central ellipsoid (earth)

n,m = Degree and order of each term

ϕ = Geocentric latitude

λ = Geocentric longitude

$\bar{C}_{nm}, \bar{S}_{nm}$ = Normalized gravitational coefficients

$\bar{P}_{nm}$ = Normalized associated Legendre functions

N = Number of terms to be used

The terms $\bar{C}_{nm}$, $\bar{S}_{nm}$, $\mu_{\oplus}$, and a are determined by the geopotential model used for evaluation (GEM 10B for STS-51). What is noteworthy here is that the terms ϕ, λ, and r are functions of the coordinates of the position vector expressed in the Earth Centered/Earth Fixed coordinate system that is tied to the geopotential model. Recognizing that the acceleration caused by this function is merely the gradient of $V_{\oplus}$, then too must acceleration be a function of position.

Recall that the intent is to apply these accelerations to two bodies in very similar orbits, thus having very similar position vectors. For proximity operations, the vector from the orbiter to the target is rarely much more than two tenths of a percent of the corresponding radius vectors. Thus the accelerations caused by spherical harmonics for the orbiter are nearly equal to those of the target. Therefor their effect on relative motion can be neglected for short propagations, corresponding to the neglect of the double summation term in Equation 5.1.

The other acceleration which we hope to neglect is that caused by drag. The drag model used for the Cowell propagator is based on the Jacchia density model, which computationally speaking, is not a trivial calculation. Drag accelerations are a function of velocity, density, and ballistic coefficient. Applying the same reasoning used relating the position vectors of chaser and target, it is argued that velocities must be nearly the same for similar orbits. Atmospheric density for

the two bodies is nearly identical due to the similar position vectors. However, the ballistic coefficient is a function of the shape and mass of a body and is thus fairly dissimilar for the orbiter and a much smaller satellite. It is precisely this difference that limits the validity of propagation without a drag acceleration. Drag effects become significant in a matter of days, thus neglecting them for more than a few orbits is not recommended.

Having justified a low fidelity propagator for relative motion prediction improves the time problem imposed by the relatively slow processor. It however does not speed the calculations to the point that they could be included. Calling a numerical integrator 80 times proved to take too long regardless of the simplicity of the acceleration term. A faster method was still required.

B. f AND g FUNCTIONS

The f and g functions, and their corresponding time derivatives, represent the basis for the standard parameterization of position and velocity along an ellipse in 3-space. They are as close as possible to analytic functions of time as can be produced for propagation in cartesian coordinates. There is one very short convergent numerical algorithm needed to evaluate the f and g functions. However, the associated function evaluations are very simple, and convergence nearly always occurs within two or three iterations.

Construction of the f(t) and g(t) functions requires an initial position vector and an initial velocity vector. As an intermediate step in the determination of f(t) and g(t), four of the cartesian elements that describe the orbit must be calculated. When time is input, a corresponding eccentric anomaly is calculated numerically, and a new position vector can be produced via [Ref. 4]

$$\begin{aligned} f(t) &= \frac{a}{r_o}\left[\cos(E(t) - E_o) - 1\right] + 1 \\ g(t) &= t + \frac{1}{n}\left[\sin(E(t) - E_o) - (E(t) - E_o)\right] \\ \vec{r}(t) &= f(t)\vec{r}_o + g(t)\vec{v}_o \end{aligned} \tag{5.2}$$

Correspondingly, velocity being the time derivative of position, a new velocity is then

$$\begin{aligned} \dot{f}(t) &= -\frac{a}{r(t)}\frac{a}{r_o} n\sin(E(t) - E_o) \\ \dot{g}(t) &= \frac{a}{r(t)}\left[\cos(E(t) - E_o) - 1\right] + 1 \\ \vec{v}(t) &= \dot{f}(t)\vec{r}_o + \dot{g}(t)\vec{v}_o \end{aligned} \tag{5.3}$$

The problem of determining future position and velocity as a function of time rests on evaluation of the right hand sides of Equations 5.2 and 5.3. Given that the inputs will be initial position and velocity vectors, whose magnitude is given by

$$r_o = \|\vec{r}_o\| \qquad v_o = \|\vec{v}_o\| \tag{5.4}$$

the specific energy of the orbit is given by

$$\mathscr{E} = \frac{v_o^2}{2} - \frac{\mu_\oplus}{r_o} \tag{5.5}$$

and the semi-major axis (a) is

$$a = -\frac{\mu_\oplus}{2\mathscr{E}} \tag{5.6}$$

The angular momentum vector and corresponding magnitude for the orbit

$$\begin{aligned} \vec{h} &= \vec{r}_o \times \vec{v}_o \\ h &= \|\vec{h}\| \end{aligned} \tag{5.7}$$

are assumed constant. The eccentricity vector, given by

$$\vec{e} = \left(\frac{v_o^2}{\mu_\oplus} - \frac{1}{r_o}\right)\vec{r}_o - \left(\frac{\vec{r}_o \cdot \vec{v}_o}{\mu_\oplus}\right)\vec{v}_o \tag{5.8}$$

points toward perigee and has magnitude

$$e = \|\vec{e}\| \tag{5.9}$$

equal to the eccentricity of the orbit. The true anomaly (ν_o) corresponding to this point is obtained from [Ref. 5]

$$\begin{aligned} \cos\nu_o &= \frac{\vec{r}_o \cdot \vec{e}}{r_o e} \\ \sin\nu_o &= \frac{\vec{h} \cdot (\vec{e} \times \vec{r}_o)}{h e r_o} \\ \tan\nu_o &= \frac{\sin\nu_o}{\cos\nu_o} \end{aligned} \tag{5.10}$$

Given the initial true anomaly, the initial eccentric anomaly (E_o) can then be found from [Ref. 1]

$$\tan\left[\frac{E_o}{2}\right] = \tan\left[\frac{\nu_o}{2}\right]\sqrt{\frac{1-e}{1+e}} \qquad (5.11)$$

It will also prove convenient to define the initial mean anomaly (M_o) via Kepler's equation

$$M_o = E_o - e\sin(E_o) \qquad (5.12)$$

The final parameter needed for Equations 5.2 and 5.3 is the mean motion (n) of the orbit, given by

$$n = \sqrt{\frac{\mu_\oplus}{a^3}} \qquad (5.13)$$

Note that the calculations represented by Equations 5.4 through 5.13 need not be performed 80 times to produce the theoretical 40 relative positions, but rather twice (once for the chaser, and once for the target). To evaluate Equations 5.2 and 5.3 for a given time (measured positively from $t=0 \Rightarrow \vec{r}_o, \vec{v}_o$) requires the mean anomaly as a function of time from

$$M(t) = M_o + n\,t \qquad (5.14)$$

and the numerical solution of Kepler's equation for the eccentric anomaly at that time

$$M(t) = E(t) - e\sin(E(t)) \qquad (5.15)$$

The solution of Equation 5.15 is found by applying the Laguerre-Conway algorithm [Ref. 1], which is an iterative root solver. Because the function evaluations are not complex, and

the orbits of interest are near circular, convergence never takes more than three iterations.

With e, a, and E(t) in hand, the magnitude of the radius as a function of time (r(t)) is

$$r(t) = a(1 - e\cos(E(t))) \qquad (5.16)$$

completing the list of necessary terms for evaluation of Equations 5.2 and 5.3 for times beyond that corresponding to the initial position and velocity.

Equations 5.14 through 5.16 are the only calculations that must be performed for each time of interest, providing a very fast means of propagation in cartesian coordinates. Specifically, this algorithm is fast enough to be performed continuously so as to provide a means of having a constant prediction of future motion.

C. FUTURE PLOTS

Keeping in mind that maneuvers are normally performed near the VBAR, this algorithm provides a means of predicting when a maneuver will be needed, as well as giving some intuitive feel for what maneuver should be performed. Figure 5.1 shows an RBAR/VBAR display screen with the Future Plot option selected. The number of points to display and interval between points are user defined options. For this simulation, 40 points at an interval of three minutes are used. Prediction begins where the curve turns from solid (history) to numbered.

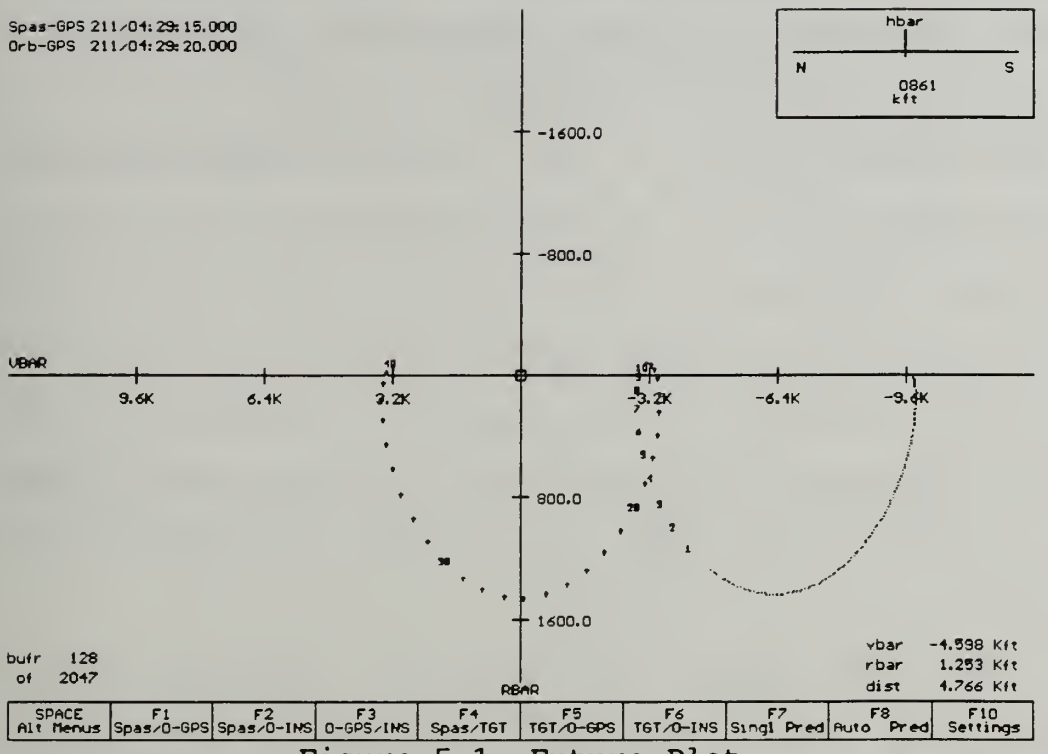

Figure 5.1 Future Plot

At 40 increments, or two hours, the orbiter will be cresting just above the VBAR. This point, or the initial predicted crest (at 11 increments), correspond to the points where maneuver should be performed to affect a rendezvous.

D. FUTURE THRUST

Given that the astronauts have some physical intuition for the type of maneuver that should be performed given a position and motion displayed on an RBAR/VBAR plot, use of the Future Thrust algorithm provides an opportunity to refine the thrusts required to perform a rendezvous. The first step is to use a Future Plot to note the time that a maneuver should be

performed. The user then initiates the Future Thrust algorithm with a predicted thrust and thrust time. Future Thrust uses the f and g functions to propagate the state vectors to that time, and then applies the input velocity change (thrust), and propagation continues. If the continuation of the Future Plot does not intersect the desired point, then the user simply modifies the input thrust. Because this algorithm is so fast, modifying the thrust settings appears to immediately vary the track displayed on the Future Plot, thus providing a means of "walking" the track onto a desired point (presumably near the target).

Velocity changes are normally expressed in LVLH coordinates, however they do not represent a derivative with respect to a moving frame. They are in fact inertial derivatives simply expressed in a convenient frame. The familiar transformation matrix ${}^{I}C^{L}$ is used to transform the velocity changes into inertial coordinates. The transformation matrix ${}^{L}C^{I}$, produced in Chapter III, consists of three orthonormal basis vectors, and therefor the inverse of ${}^{L}C^{I}$ is merely its transpose.

The user inputs to this algorithm are a time to affect the thrust, which is set on a separate screen, and the thrusts applied at that time, controlled via keys when the RBAR/VBAR screen is active. Figure 5.2 shows a Future Thrust screen created during the same simulation that created Figure 5.1. The user has chosen to perform the thrust at the first VBAR

crossing. The inner numbered curve represents the predicted path after the thrust, while the outer numbered curve is the

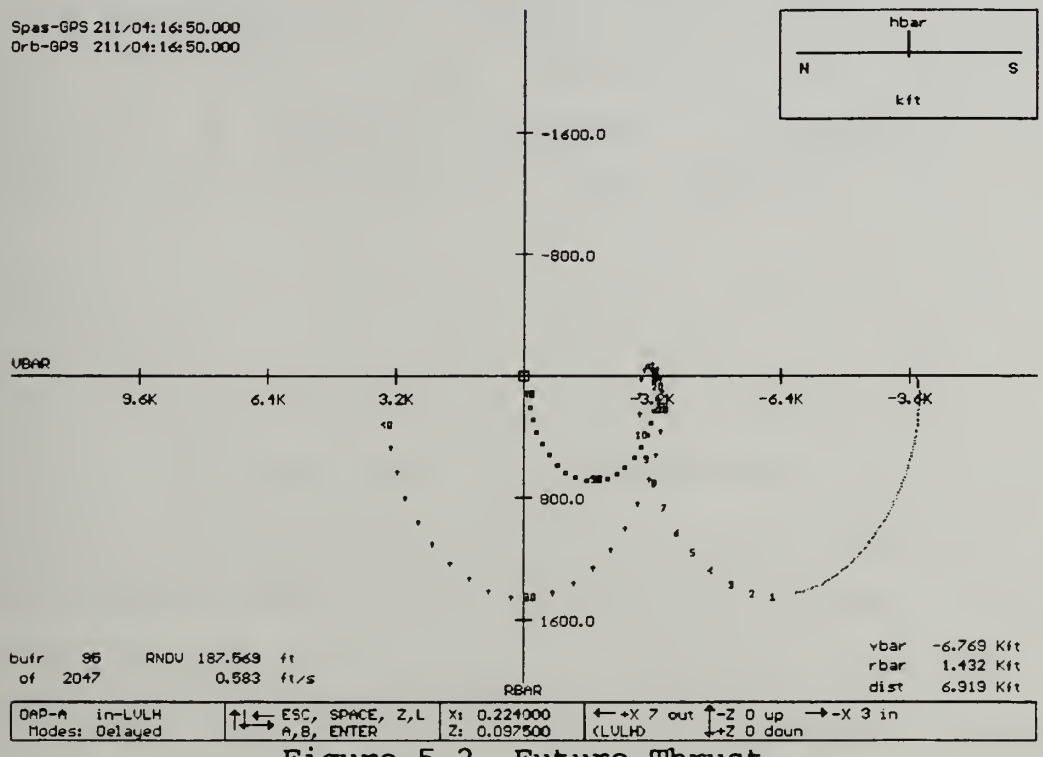

Figure 5.2 Future Thrust

predicted path if the thrust command is ignored. The LVLH thrusts in the "x" and "z" direction are displayed within the menu bar. These correspond to values to be programmed into the orbiter's control system for activation at the predicted time.

E. RENDEZVOUS PREDICTION

In the purest sense, Future Thrust does not represent a rendezvous algorithm. No rendezvous solution is provided. However, given the intuition of the astronauts, this algorithm provides the opportunity to evaluate the effectiveness of proposed thrusts before they are performed. Operationally,

thrust commands are calculated on the ground and then uplinked. The crew merely performs the commanded maneuvers. These algorithms provided a means of viewing the results of ground directed thrusts on orbit. To this end, these algorithms enhance the situational awareness of the crew, and provide an independent means of validating ground commands.

VI. RENDEZVOUS SOLUTION

The methods discussed in this chapter represent one way to predict the velocity changes needed to affect a rendezvous. Although it was possible to stretch the requirements of DTO 700 to provide information that increased the situational awareness of the flight crew, rendezvous prediction was in no way addressed in DTO 700. Essentially, as long as no information was provided that could be used to *control* the orbiter, experimentation with algorithms was permitted. Producing the thrusts required to affect a rendezvous are indeed control inputs. To be considered for flight, a *control* algorithm has to undergo strenuous testing. Given there was insufficient time for such testing, the algorithms presented in this chapter were not included in the *primary* executable that flew on STS-51. The rendezvous initiation portion of this algorithm was, however, available in a backup executable, should a need for its output have arisen.

A. LINEARIZED EQUATIONS OF RELATIVE MOTION

1. Coordinate System

This treatment is taken from Reference 1. To express the equations of relative motion, a convenient coordinate system must be used. The author has chosen a system as yet unaddressed in this thesis. It is essentially an alternate way

of creating an LVLH frame, which is an orthonormal frame tied to orbital motion. This frame is not coincident with the previously derived LVLH coordinate system. It will again be denoted with an "L" superscript, but care must be taken in avoiding the mistake that this transformation matrix is equivalent to that derived in previous chapters.

For this new coordinate system, the positive "y" axis is defined by the normalized radius vector of the target

$$\hat{y} = \frac{\vec{r}_t}{\|\vec{r}_t\|} \tag{6.1}$$

while the "z" axis is defined by the normalized angular momentum vector of the target

$$\hat{z} = \frac{\vec{H}_t}{\|\vec{H}_t\|} \tag{6.2}$$

forcing the "x" axis to be given by

$$\hat{x} = \hat{y} \times \hat{z} \tag{6.3}$$

Note that the "x" axis now points opposite the direction of travel. These orthonormal vectors define the columns of the transformation matrix from inertial to LVLH coordinates

$${}^{L}C^{I} = [\hat{x}\ \hat{y}\ \hat{z}] \tag{6.4}$$

2. Equations of Motion

The linearized equations of relative motion make use of two significant simplifications in their derivation. First,

the angular displacement between the target and chaser is assumed to be small. Recall a similar assumption was made in justifying the use of the f and g functions in Chapter V. Secondly, the equations are based upon the assumption that the target's orbit is circular. This second assumption is more restrictive than the treatment given in Chapter V. Shuttle orbits are in fact nearly circular. However, the assumption of a circular orbit does introduce a source of error.

The equations of motion are expressed in terms of initial relative position and velocity components as a function of time. Relative position is

$$
\begin{aligned}
x &= x_o + 2\frac{\dot{y}_o}{\omega}(1 - \cos\omega t) \\
&\quad + \left(4\frac{\dot{x}_o}{\omega} - 6y_o\right)\sin\omega t + (6\omega y_o - 3\dot{x}_o)t \\
y &= 4y_o - 2\frac{\dot{x}_o}{\omega} + \left(2\frac{\dot{x}_o}{\omega} - 3y_o\right)\cos\omega t + \frac{\dot{y}_o}{\omega}\sin\omega t \\
z &= \frac{\dot{z}_o}{\omega}\sin\omega t + z_o\cos\omega t
\end{aligned}
\tag{6.5}
$$

and relative velocity is given by

$$
\begin{aligned}
\dot{x} &= 2\dot{y}_o\sin\omega t + (4\dot{x}_o - 6\omega y_o)\cos\omega t + 6\omega y_o - 3\dot{x}_o \\
\dot{y} &= (3\omega y_o - 2\dot{x}_o)\sin\omega t + \dot{y}_o\cos\omega t \\
\dot{z} &= \dot{z}_o\cos\omega t - z_o\sin\omega t
\end{aligned}
\tag{6.6}
$$

The term ω in these equations is the orbital angular rate of the assumed circular orbit of the target. The corresponding term for elliptical orbits is the mean motion

(n). Without loss of generality, the mean motion for the target will be used instead of ω within the following algorithms. Note that velocity terms in Equations 6.5 and 6.6 are truly derivatives with respect to the non-inertial frame LVLH. In addition, the simplifying assumptions result in the "z" or "out of plane" equations becoming uncoupled.

For a given initial relative position and prescribed time, solving the homogeneous form of Equations 6.5 for the initial relative velocity terms ($\dot{x}_o$, $\dot{y}_o$, and $\dot{z}_o$) produces the required initial relative velocity to begin a rendezvous[3].

$$\frac{\dot{x}_r}{\omega} = \frac{x_o \sin\omega t + y_o [6\omega t \sin\omega t - 14(1 - \cos\omega t)]}{3\omega t \sin\omega t - 8(1 - \cos\omega t)}$$

$$\frac{\dot{y}_r}{\omega} = \frac{x_o + \left(4\frac{\dot{x}_r}{\omega} - 6y_o\right)\sin\omega t + (6\omega y_o - 3\dot{x}_r)t}{-2(1 - \cos\omega t)} \qquad (6.7)$$

$$\frac{\dot{z}_r}{\omega} = \frac{-z_o}{\tan\omega t}$$

With these velocities, the initial relative position, and the prescribed rendezvous time; Equation 6.6 can be solved for the terminal relative velocity which must then be negated in order to complete the rendezvous.

---

[3]The "y" component of Equation 6.7 differs from that presented in reference 1. The author's equation could not be reproduced, and furthermore did not produce a rendezvous solution during simulation.

B. ALGORITHM

1. Rendezvous Initiation Thrust

As in all algorithms presented in this thesis, the input is a pair of concurrent state vectors. Also required is an input time desired to perform the rendezvous. Since Equations 6.6 and 6.7 do not require the chaser to be on the VBAR, times that are integral multiples of orbital periods should be avoided. This is because at these times the spacecraft is condemned to pass through the same point again, and if that point is at the wrong altitude, the algorithm will try to produce huge changes in radial velocity in order to obtain the desired altitude.

Given a rendezvous time, parameters used in Equation 6.7 must be determined. Recall the position and velocity vectors for chaser and target are provided as inputs. Thus, the vector from target to chaser ($\vec{r}_d$) is given by

$$\left[\vec{r}_d = \vec{r}_c - \vec{r}_t \right]_I \tag{6.8}$$

The subscript "I" is used as a reminder that the equation is expressed in inertial coordinates. The following equations contain inertial and non-inertial derivatives. Vector equations which do not perform a coordinate transformation will be subscripted to denote which coordinate system the equation is expressed in.

Construction of the matrix ${}^{L}C^{I}$ as a function of the target state vector is outlined in Equations 6.1 through 6.4. The difference vector is expressed in the new coordinate system by

$$\left[\vec{r}_{\beta}\right]_{L} = {}^{L}C^{I}\left[\vec{r}_{\beta}\right]_{I} \tag{6.9}$$

where the components of the left hand side vector of Equation 6.9 represent the variables x_o, y_o, and z_o of Equation 6.7.

As previously noted, the orbital angular rate (ω) is represented using the more general concept of mean motion (n), which in turn requires a value for the semi-major axis of the target orbit. Given target position and velocity vectors

$$r_t = \|\vec{r}_t\| \qquad\qquad v_t = \|\vec{v}_t\| \tag{6.10}$$

the relationship between radius, speed and semi-major axis

$$v_t = \sqrt{\mu_{\oplus}\left(\frac{2}{r_t} - \frac{1}{a_t}\right)} \tag{6.11}$$

can be solved for the semi-major axis

$$a_t = \frac{r_t \mu_{\oplus}}{2\mu_{\oplus} - r_t v_t^2} \tag{6.12}$$

The mean motion (or orbital angular rate in this case) is finally given by

$$\omega = n = \sqrt{\frac{a_c^3}{\mu_\oplus}} \tag{6.13}$$

providing the last parameter required to solve Equation 6.7 for the initial relative velocity necessary to start the rendezvous.

The solutions produced by Equation 6.7 are clearly components of a velocity vector. The problem is to relate this relative velocity to a desired change in the chaser velocity vector. The vector created by Equation 6.7 is the time derivative of the difference vector created with Equations 6.8 and 6.9. The crucial point is that it is a derivative with respect to the LVLH frame and not with respect to inertial space. To relate derivatives taken with respect to two different frames, the relationship

$$\frac{^A d\vec{r}}{dt} = \frac{^B d\vec{r}}{dt} + {}^A\vec{\omega}^B \times \vec{r} \tag{6.14}$$

is used, where the term ${}^A\vec{\omega}^B$ is read "the rotation rate of coordinate system B with respect to coordinate system A". In terms of the coordinate systems used,

$$\left[\frac{^I d\vec{r}_d}{dt} = \frac{^L d\vec{r}_d}{dt} + {}^I\vec{\omega}^L \times \vec{r}_d \right]_L \tag{6.15}$$

where the derivative on the right hand side is with respect to LVLH (superscript "L") and is the output of Equation 6.7, and ${}^I\vec{\omega}^L$ expressed in LVLH coordinates is simply

$$ {}^{I}\vec{\omega}^{L} = \begin{bmatrix} 0 \\ 0 \\ \omega \end{bmatrix}_{L} = \begin{bmatrix} 0 \\ 0 \\ n \end{bmatrix}_{L} \qquad (6.16) $$

Equation 6.15 provides the means to evaluate the time derivative of the difference vector with respect to inertial space. However, note the left hand side of Equation 6.15 represents the difference of the velocity vectors.

$$ \left[{}^{I}\frac{d\vec{r}_{d}}{dt} = {}^{I}\frac{d\vec{r}_{c}}{dt} - {}^{I}\frac{d\vec{r}_{t}}{dt} \right]_{I} \qquad (6.17) $$

$$ = \vec{v}_{c} - \vec{v}_{t} $$

Equation 6.15 is expressed in LVLH coordinates, while Equation 6.17 is expressed in inertial coordinates. To transform to inertial coordinates, the results of Equation 6.15 must be left multiplied by the inverse of ${}^{L}C^{I}$ (which is its transpose)

$$ \left[{}^{I}\frac{d\vec{r}_{d}}{dt} \right]_{I} = {}^{I}C^{L} \left[{}^{I}\frac{d\vec{r}_{d}}{dt} \right]_{L} \qquad (6.18) $$

providing the means to solve Equation 6.17 for the desired chaser velocity in inertial coordinates

$$ \left[\vec{v}_{c} = {}^{I}\frac{d\vec{r}_{d}}{dt} + \vec{v}_{t} \right]_{I} \qquad (6.19) $$

The chaser velocity given by Equation 6.19 is the desired velocity expressed in inertial coordinates needed to affect a rendezvous. Recall a value for $\vec{\mathbf{v}}_{\mathbf{c}}$ was input to this

algorithm. The input value represents the current chaser velocity, while Equation 6.19 gives the desired chaser velocity. Differencing these produces the desired velocity change to affect the rendezvous

$$\Delta\vec{v}_c = \left[\vec{v}_c\right]_{new} - \left[\vec{v}_c\right]_{old} \qquad (6.20)$$

which, though produced in inertial coordinates, can be expressed in any convenient coordinate system given the transformation matrices created in prior chapters.

2. **Rendezvous Termination Thrust**

Although initially tested with the software designed for STS-51, this portion of the algorithm has since been removed, and did not fly in any form as part of DTO 700. As will be shown, the Δv produced by this algorithm is based on information as old as the chosen time to rendezvous. Said another way, as a rendezvous proceeds, more timely information becomes available, thus antiquating this solution.

To produce the desired rendezvous termination thrust, the initial relative velocity for the rendezvous was found from Equation 6.7. The results of Equations 6.7, 6.8, 6.9, 6.13, and the desired time to rendezvous, are all of the inputs needed to solve Equation 6.6 for the relative velocity at the end of the maneuver. Applying Equations 6.15 through 6.19 to the output of Equation 6.6 produces the inertial velocity at the end of the rendezvous ellipse, which

corresponds to the old chaser velocity vector of Equation 6.20. Clearly, the desired new chaser velocity should be identical to the target velocity. It may be assumed that the position vector difference is zero, since that is how the algorithm began.

The target state vector must be propagated through the desired time to rendezvous, in order to produce the transformation matrix of Equation 6.18, and the target velocity vector of Equations 6.19 and 6.20, before applying Equations 6.15 through 6.20.

During the testing of this algorithm, the Cowell propagator was used to produce the target state vector at rendezvous termination. However, since the solution of the equations of relative motion, and the solutions produced by the f and g functions are based on many of the same assumptions, propagation with the f and g functions were considered equally valid.

C. APPLICATION

Rendezvous maneuvers for the Shuttle are typically performed on the VBAR. However, should the need arise, this algorithm is not so constrained. Figure 6.1 shows the display screen associated with the algorithm. Because inclusion of this algorithm was a relatively low priority item, the actual thrusts computed are displayed on a separate screen. In this display, the algorithm is coupled with the Future Thrust

algorithm. The solution for the desired Δv is directly input to the Future Thrust algorithm with zero time delay. The 20 time increments shown in Figure 6.1 correspond to a one hour "look ahead" for Future Plot. Coincidently, one hour is the

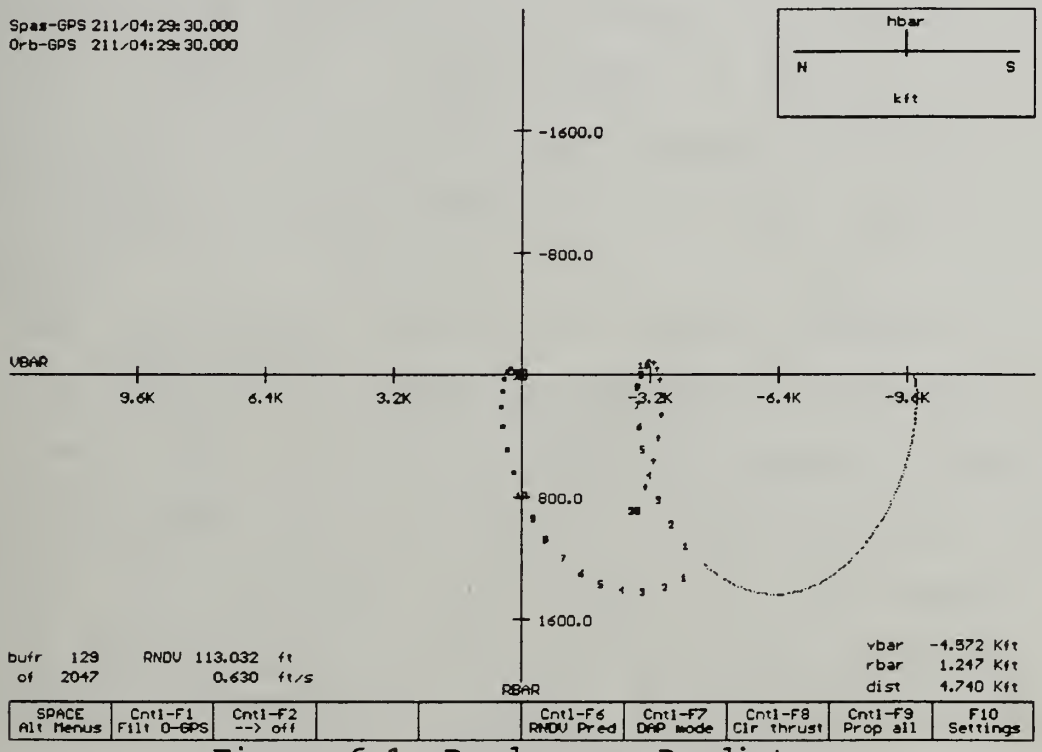

Figure 6.1 Rendezvous Predictor

prescribed time input to the rendezvous initiation thrust algorithm. The figure shows that it will take 30 minutes to reach the VBAR. With the standard methods of maneuvering on the VBAR, it would then take another orbital period (≈90 minutes) to affect a rendezvous. The numbered curve that shows the abrupt change in direction results from the rendezvous initiation thrust algorithm, and demonstrates a means for rapid rendezvous not available with current methods.

VII. RENDEZVOUS SOLUTION (REVISITED)

The previous chapters address the theory behind the algorithms that resided in the NPS software that flew as part of DTO 700 on STS-51. However, this software was not designed such that it could only support STS-51, but rather it can be molded to provide operationally significant information given any source of state vectors. The crew of STS-60, scheduled for a December launch, is currently planning on using some version of this software during their mission. Furthermore, much of the functionality of the NPS software is coincident with that of another program more commonly used by NASA. There is currently much interest in merging the two programs, producing an *ultimate* rendezvous and proximity operations tool. Because the software created by the NPS team is still very much in the spotlight, continuing to address the problems of rendezvous and proximity operations is prudent.

This chapter will readdress the problem of producing the necessary thrusts required to affect a rendezvous, given a prescribed rendezvous time. The treatment is taken from *Fundamentals of Astrodynamics* [Ref. 6], and is more general than the methods of Chapter VI. The algorithm is based on the use of the f and g functions, thus benefiting from the assertions to their validity in proximity operations put forward in Chapter V.

A. THE GAUSS PROBLEM

The Gauss Problem is a general term associated with the problem of orbit determination from observations at specific times. In this chapter, the problem of orbit determination given two position vectors and a time between them will be addressed. The brilliant German mathematician, Carl Friedrich Gauss, did not have the luxury of being presented complete position vectors, but rather had only the right ascensions and declinations of three observations. The methods used by Gauss are, however, equally valid for the more simply stated problem.

1. Formulation

Inputs for the Gauss Problem are two position vectors and a time. However, as with all algorithms in this thesis, the information provided consists of two concurrent state vectors, and a desired time to rendezvous. To formulate the Gauss Problem, the position vector of the chaser may be used directly. However, the second point in the *rendezvous ellipse* will be a point occupied by the target after the time to rendezvous has elapsed. In other words, the target state vector must be propagated through the time to rendezvous to produce the second position vector. Both *rendezvous ellipse* and the future target position will be produced with the f and g functions, neglecting perturbing accelerations, invoking the

assertions of Chapter V as to the *relative* accuracy of the solution.

The two input vectors for the Gauss Problem are

$$\begin{matrix} \vec{r}_1 = \vec{r}_c & \vec{r}_2 = \vec{r}_t \\ \vec{v}_c & \vec{v}_t \end{matrix} \qquad (7.1)$$

where the "t" subscripted vectors are those for the target at the time of rendezvous. The velocity vectors are not required for the Gauss Problem, however they will be required later to produce the thrusts necessary to begin and terminate the rendezvous.

2. Solution

Before addressing the solution to the Gauss Problem, the difference in true anomalies ($\Delta\nu$) for the two position vectors is needed. Calculation of $\Delta\nu$ using an inner product is insufficient, since it does not give the sign of $\Delta\nu$, which is also important. Given the position vectors and magnitudes

$$r_1 = \|\vec{r}_1\| \qquad r_2 = \|\vec{r}_2\| \qquad (7.2)$$

the difference of true anomalies is found from

$$\begin{aligned} \cos(\Delta\nu) &= \frac{\vec{r}_1 \cdot \vec{r}_2}{r_1 r_2} \\ \sin(\Delta\nu) &= \left(\frac{\vec{r}_1 \times \vec{r}_2}{r_1 r_2}\right) \cdot \left(\frac{\vec{r}_t \times \vec{v}_t}{\|\vec{r}_t \times \vec{v}_t\|}\right) \\ \tan(\Delta\nu) &= \frac{\sin(\Delta\nu)}{\cos(\Delta\nu)} \end{aligned} \qquad (7.3)$$

The second term in the sine equation is the normalized angular momentum vector for the target's orbit. Ideally, the angular momentum vector from the transfer ellipse should be used. Since it is unavailable, this term serves as a good approximation for similar orbits.

As previously stated, this algorithm is dependent on the use of the f and g functions introduced in Chapter V. The form used to calculate the values for f, g, and their derivatives varies from that used previously. This is due to the iterative method used in solving the problem. It will prove convenient to choose the semi-latus rectum (p) as the variable to iterate on, forcing an additional form of the f and g functions.

$$
\begin{aligned}
f &= 1 - \frac{r_2}{p}(1 - \cos\Delta\nu) = 1 - \frac{a}{r_1}(1 - \cos\Delta E) \\
g &= \frac{r_1 r_2 \sin\Delta\nu}{\sqrt{\mu p}} = t - \sqrt{\frac{a^3}{\mu}}(\Delta E - \sin\Delta E) \\
\dot{f} &= \sqrt{\frac{\mu}{p}}\tan\left(\frac{\Delta\nu}{2}\right)\left(\frac{1 - \cos\Delta\nu}{p} - \frac{1}{r_1} - \frac{1}{r_2}\right) = \frac{-\sqrt{\mu a}}{r_1 r_2}\sin\Delta E \\
\dot{g} &= 1 - \frac{r_1}{p}(1 - \cos\Delta\nu) = 1 - \frac{a}{r_2}(1 - \cos\Delta E)
\end{aligned}
\qquad (7.4)
$$

Recall, the f and g functions define the rendezvous ellipse by

$$\vec{r}_2 = f\vec{r}_1 + g\vec{v}_1$$
$$\vec{v}_2 = \dot{f}\vec{r}_1 + \dot{g}\vec{v}_1 \qquad (7.5)$$

Since the position vectors in Equation 7.5 are known, the Gauss problem is reduced to solving for the scalars f, g, $\dot{f}$, and $\dot{g}$.

Since it can be shown that one of the expressions in Equations 7.4 is dependent, we have three equations in seven variables (r_1, r_2, $\Delta\nu$, t, p, a, and ΔE). However, four of these variables are known (r_1, r_2, $\Delta\nu$, and t). The problem is then to solve three transcendental equations in three unknowns. Many iterative methods are available, however a particularly elegant method is taken from Reference 6.

3. p-Iteration Method

There are many advantages to the p-iteration method. One of particular importance for numerical computing, is that this method allows the implementation of a Newton iteration for convergence. The elegance of this method stems from the fact that the semi-latus rectum (p) can be expressed in terms of three of the known variables and only one of the unknowns.

$$p = \frac{r_1 r_2 (1 - \cos\Delta\nu)}{r_1 + r_2 - 2\sqrt{r_1 r_2}\cos\frac{\Delta\nu}{2}\cos\frac{\Delta E}{2}} \qquad (7.6)$$

Likewise, the semi-major (a) axis can be expressed in terms of the single variable p by

$$a = \frac{MKp}{(2M - L^2)p^2 + 2KLp - K^2} \tag{7.7}$$

where the parameters M, K, and L are directly calculated by

$$\begin{aligned} K &= r_1 r_2 (1 - \cos\Delta v) \\ L &= r_1 + r_2 \\ M &= r_1 r_2 (1 + \cos\Delta v) \end{aligned} \tag{7.8}$$

For a guessed value of p, the formulas presented thus far provide the means of solving for t in the second formula of Equations 7.4 (this will be outlined in summary). The question then becomes; how to pick an initial value for p, and how to update p between iterations.

The methods for guessing an initial p presented by Bate, Mueller and White are much more general than is required for the present problem. A transfer ellipse similar to the target and chaser orbits is required. These orbits are nearly circular (e = 0). Specifically, since semi-latus rectum is given by

$$p = a(1 - e^2) \tag{7.9}$$

an average of the two radii, conveniently given by

$$p_c = \frac{r_1 + r_2}{2} = \frac{L}{2} \tag{7.10}$$

represents an outstanding initial value for semi-latus rectum.

As previously mentioned, the p-iteration method provides a means of using a Newton iteration for updating the values of p. Recall that a guessed value of p eventually

produces a value for time (t_n) in Equation 7.4. To update p, the Newton iteration

$$p_{n+1} = p_n + \frac{t - t_n}{\left[\frac{dt}{dp}\right]_{p=p_n}} \tag{7.11}$$

requires the evaluation of the derivative of time with respect to semi-latus rectum at the guessed value of p. A straightforward calculation provides this derivative as a function of variables either known or calculated during the iteration.

$$\frac{dt}{dp} = \frac{-g}{2p} - \frac{3}{2}a(t-g)\left(\frac{K^2+(2M-L^2)p^2}{MKp^2}\right) + \sqrt{\frac{a^3}{\mu}}\frac{2K\sin\Delta E}{p(K-Lp)} \tag{7.12}$$

Equation 7.12 represents the derivative corresponding to an elliptical transfer orbit. A hyperbolic solution may also be possible (requiring a different formula), but such a transfer orbit would require an inordinate amount of energy to be expended. Therefor, only Equation 7.12 will be considered.

B. ALGORITHM

Given concurrent state vectors for the target and chaser, and a desired time to rendezvous, the following steps will provide a rendezvous ellipse based on the two body solution as embodied in the f and g functions:

1. Use the f and g algorithms discussed in Chapter V to propagate the target state vector to the time of rendezvous. This state vector represents position 2 for

the Gauss Problem. Position 1 is represented by the current chaser state vector.

2. Calculate the difference in true anomalies with Equation 7.3.

3. Calculate the parameters K, L, and M with Equation 7.8.

4. Guess an initial value for the semi-latus rectum with Equation 7.10, and initialize t_n at zero.

While $|t-t_n| > \varepsilon$ do steps 5 through 9 (t = time to rendezvous)

5. Calculate f, g, and $\dot{f}$ with the Δv formulation of Equation 7.4.

6. Rewriting the ΔE formulation of Equation 7.4

$$\cos\Delta E = 1 - \frac{r_1}{a}(1 - f)$$

$$\sin\Delta E = \frac{-r_1 r_2 \dot{f}}{\sqrt{\mu a}} \qquad (7.13)$$

$$\tan\Delta E = \frac{\sin\Delta E}{\cos\Delta E}$$

solve for ΔE.

7. Using the ΔE formulation for g in Equation 7.4, solve for t_n.

8. Calculate dt/dp from Equation 7.12 using t_n and p_n.

9. Produce the next guess for p with Equation 7.11.

Assuming $|t-t_n|$ is adequately small, the last values for p, a, and ΔE are now acceptable.

10. Calculate f, g, $\dot{f}$, and $\dot{g}$ with Equation 7.4.

11. Solve for $\vec{v}_1$ in the first formula of Equation 7.5.

12. Solve for $\vec{v}_2$ in the second formula of Equation 7.5.

13. Solve for the rendezvous initiation thrust ($\Delta\vec{v}_1$) via

$$\Delta\vec{v}_1 = \vec{v}_1 - \vec{v}_c \qquad (7.14)$$

14. Solve for the rendezvous termination thrust ($\Delta\vec{v}_2$) via

$$\Delta\vec{v}_2 = \vec{v}_t - \vec{v}_2 \qquad (7.15)$$

C. ADVANTAGES

Compared to the rendezvous solution produced with the linearized equations of relative motion, this algorithm has two very distinct advantages. First this algorithm is clearly more general. There is no near circular orbit assumption, and the algorithm is valid regardless of the relative distance between target and chaser. The quality of the solution will decrease slightly because of accuracy lost in not accounting for perturbing accelerations, however this method will still get the chaser in the neighborhood of the target[4].

The second advantage is the ability to bias the algorithm to a particular size orbit. The linearized equations of relative motion offer no means of sizing an orbit. Both methods produce an ellipse constrained with two positions and a time. However, there is not a unique solution to this problem. For example, assume the time constraint is two hours. The rendezvous ellipse may be one that passes through the rendezvous point once before rendezvous, or it may be the more

---

[4]The method of determining Δv in Equation 7.3 does require the orbits to be nearly coplaner.

eccentric ellipse that just reaches the rendezvous point at the two hour point. The linearized equations of motion offer no convenient means of specifying which ellipse is desired. With the algorithm presented in this chapter, the opportunity to modify the values for Δv at step 2, and ΔE at step 6 is available. For example, a nominal period is roughly 90 minutes. If two hours is chosen as the time to rendezvous, Δv and ΔE may be increased by a factor of 2π to allow for the rendezvous during the second orbit of the rendezvous ellipse. This method of biasing the size of the rendezvous ellipse translates into smaller thrusts, and a rendezvous that "walks" toward the target.

Figure 7.1 shows the results of a simulation for which the time to rendezvous was four hours. Since this time lies between the nominal two and three orbit times, a factor of 4π was added to Δv and ΔE at the appropriate steps. The dotted curve is the result of a posigrade separation maneuver. When the chaser reached the VBAR, the rendezvous was initiated. The smooth curve is the path traveled along the rendezvous ellipse. The simulation was run with a high fidelity Cowell propagator, while the rendezvous solutions are produced with the f and g functions. Although the rendezvous was initiated over 6,400 feet away, the rendezvous termination thrust occurred at a point less than 20 feet from the target.

In a very practical sense, the two time constrained rendezvous algorithms presented apply directly to a rendezvous

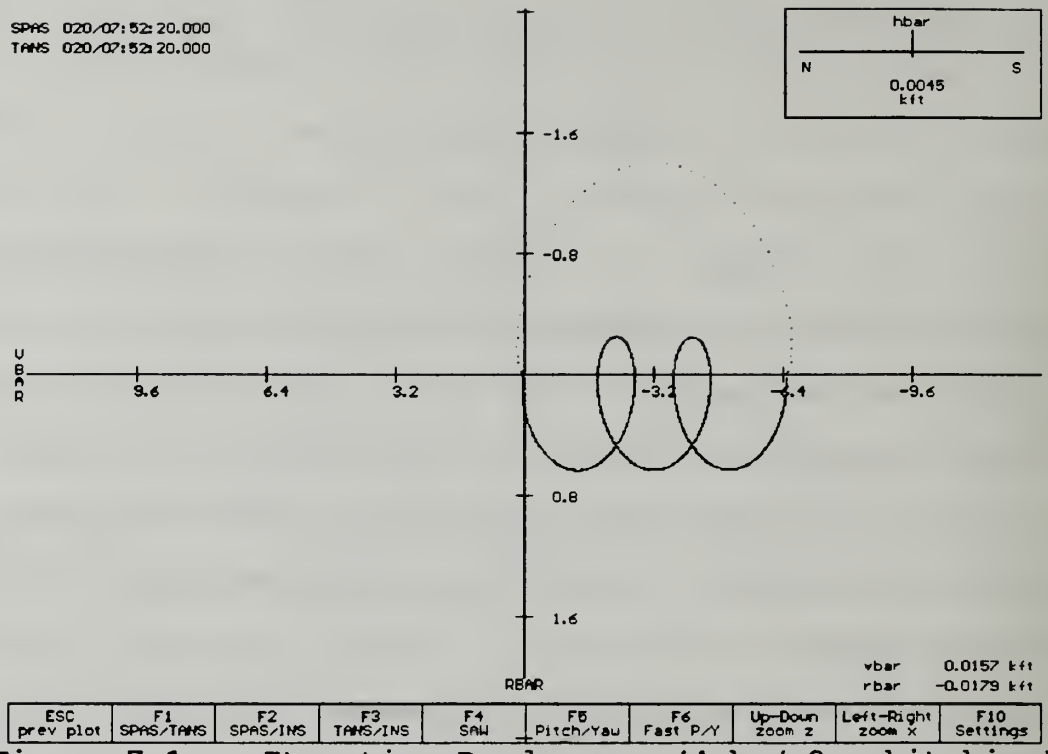

Figure 7.1 p-Iteration Rendezvous (4 hr/+2 orbit bias)

method currently employed by NASA. Current rendezvous techniques [Ref. 7] are initiated by maneuvers on the VBAR which amount to phase corrections for spacecraft in the same orbit with a phase difference. The first direct rendezvous targeting is performed on the "bounce" prior to passing the target, and initiates an angularly constrained rendezvous with a point approximately 400 feet from the target. While on this rendezvous ellipse there are maneuvers known as Midcourse Corrections (MC's) performed to refine the 400 foot point. Translating the angular constraint to a time constraint, and constraining the time at the 400 foot point provides an ideal opportunity for the employment of these algorithms.

VIII. POST FLIGHT DATA

As previously mentioned, with the exception of the state vectors provided by the TANS GPS receiver, all information used by the algorithms presented in this thesis, was stripped from the downlinked data stream. Mr. Tom Silva, the gentleman responsible for the software that did the stripping, also had the foresight to record the downlinked data at the Johnson Space Center. He then designed software that provided the means to replay this recorded data through the flight software. An exhaustive analysis of the NPS software with recorded flight data would take months, and very likely will be the subject of future theses. This chapter will concentrate on the periods just before and after the rendezvous with SPAS highlighting the usefulness of these algorithms as an on orbit, and post-flight debriefing tool.

A. ON ORBIT

1. State Vector Quality

A common phrase among computer programmers is "Garbage in, garbage out." This phrase is entirely applicable to the NPS software. The algorithms accept state vector inputs. However, if the input state vectors do not accurately represent the position and velocity of the bodies of interest, the solution produced by any of these algorithms will be

equally flawed. The primary state vectors of interest are those produced by the computers of the orbiter and SPAS. Both of these involve propagation, and thus accrue error. However, as the orbiter approaches the target, a KU band radar aboard the orbiter illuminates the target producing very accurate *relative* position and velocity information. Within the orbiter's computers, the orbiter state vector is slaved to match the relative information produced by the KU band radar. Although the true positions and velocities of the bodies may not necessarily be contained in their respective state vectors, the errors have become identical, producing the ideal scenario for the relative motion algorithms contained in the NPS software.

2. Terminal Initiation

The Terminal Initiation (TI) burn marks the beginning of a direct rendezvous using standard NASA rendezvous procedures [Ref. 7]. Prior to TI, the chaser is "bouncing" beneath the VBAR toward the target. The "bounces" are designed such that if no maneuvers are performed, the chaser will pass harmlessly beneath the target. The TI burn is performed at the last apogee prior to passing beneath the target, and is designed to begin a rendezvous that will be complete in 320°.

The constraints imposed by NASA on the various maneuvers performed during rendezvous are complicated and involve the consideration of much more information than is

available to the NPS software. However, NASA is equally hampered by the problem of inaccurate state vectors used for calculation. The calculations for the TI burn are performed during the orbit prior to TI by ground controllers, and are then uplinked to the crew of the orbiter.

Figure 8.1 shows a Future Thrust display screen at about the same time the TI burn parameters become available. The inner numbered curve shows that the orbiter will pass

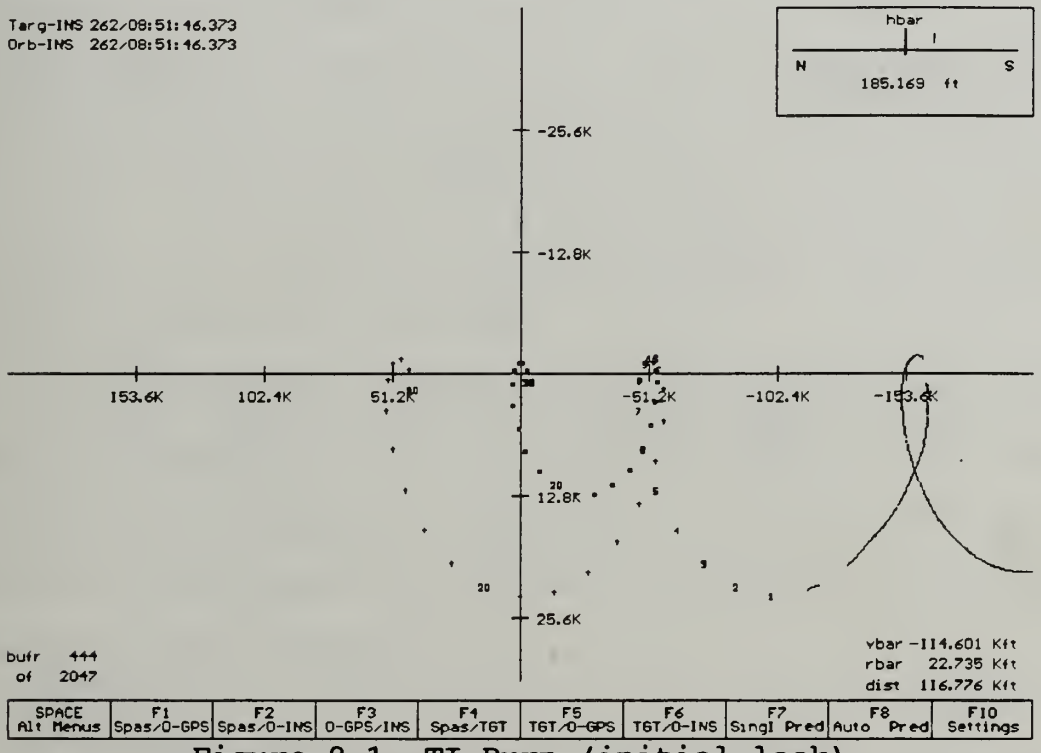

Figure 8.1 TI Burn (initial look)

slightly ahead of the target, which is exactly the desired flight path.

Leaving the Future Thrust option active as the orbiter approaches TI begins to tell a different story. As the orbiter

gets closer to the target, the orbiter state vector becomes slaved to the target state vector via KU band radar information. As the relative error between the state vectors decreases, the Future Thrust algorithm begins to show that the orbiter will not even reach the RBAR as the result of the TI burn. Figure 8.2 shows a Future Thrust display screen just prior to TI. The operator of this software, Dr. James Newman, had the ability at this point to incrementally vary the TI burn parameters so as to "walk" the predicted trajectory

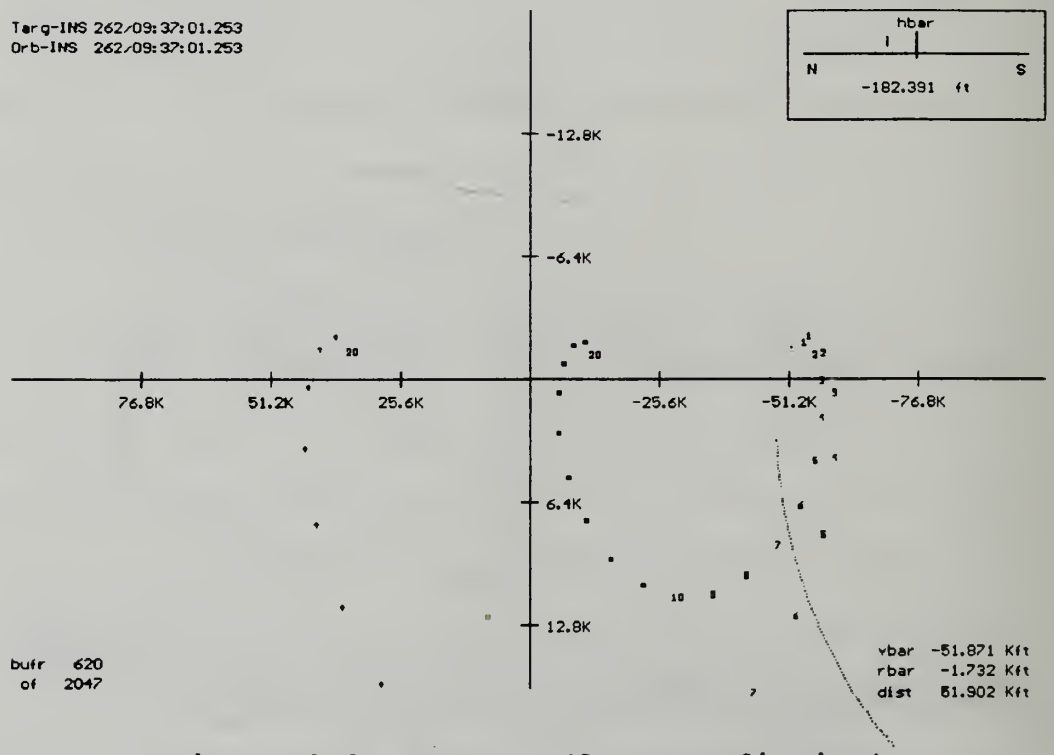

Figure 8.2 TI Burn (last prediction)

forward of the target. However, considering the experimental, and entirely unproven nature of the NPS software, this would

probably not have been a tremendously wise choice, in spite of it's availability.

Armed with the results of Figure 8.2, and the Future Plot screens produced after performance of the commanded TI burn which confirmed a trajectory that fell short of the target, the crew of STS-51 was prepared for the scenario that followed. Typically, four maneuvers, known as Midcourse Corrections (MC), are planned to follow the TI burn. These maneuvers are intended to be much smaller than the TI burn, and are designed to "sweeten" the rendezvous solution. Aware that the trajectory following the TI burn would fall short of the target (having run the Future Thrust and Future Plot algorithms), the relatively large MC commands that followed came as no surprise to the crew.

Even if the Future Thrust algorithms had been completely neglected, a Future Plot showing the short trajectory following the TI burn served to greatly enhance the situational awareness of the crew in a scenario of the type shown with this rendezvous.

B. DEBRIEFING

As with the flight of any aircraft, often more is learned after the flight than during the flight. The capability of playing back the recorded data through the flight software gives the crew the ability to view crucial moments during the

flight, quantify the actions they took, and critically evaluate those actions in terms of their results.

Following the MC burns, as the orbiter was approaching the VBAR, Captain Frank Culbertson (Mission Commander) performed a series of small maneuvers in an apparent effort to control his closure rate on the target[5]. Figure 8.3 shows the serpentine trajectory produced by the series of maneuvers and the resulting 400 ft intercept of the VBAR. The 400 ft point

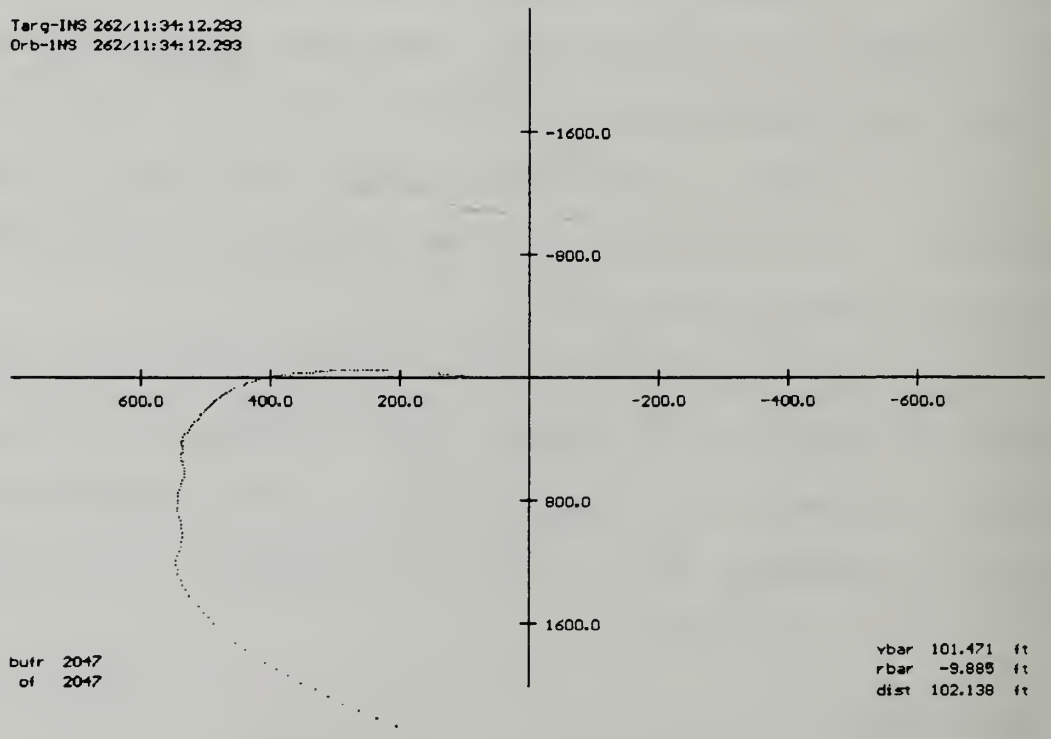

Figure 8.3 Nearing Rendezvous

[5]This conclusion was reached from viewing a video tape of the crew during the rendezvous. A comprehensive, face to face, debrief would be required to accurately assess the intentions of Captain Culbertson.

is exactly the desired VBAR crossing enroute to rendezvous. The question becomes; "Was the serpentine path necessary?"

Giving Captain Culbertson the ability to replay this portion of the flight with access to Future Plot and Future Thrust algorithms provides a definitive means of answering the question. I believe the answer to be yes, when the issue of closure rate is considered. However, Captain Culbertson may be able to shed more light on the subject in debrief.

Providing the astronauts the ability to replay portions of the flight not only enhances their ability to handle a similar scenario in the future, but could also help standardize the actions required for a given scenario.

C. GPS ACCURACY

Because the primary goal of the NPS software was to produce sawtooth plots to afford the crew a "real time" means of evaluating GPS data, some discussion in post-flight is warranted. Unfortunately, the data produced by the TANS GPS receiver was not part of the downlinked data stream, and is not available at the time of writing. Reference 2, when published, will have a detailed analysis of the TANS GPS data, although no connection with the NPS software is intended.

As mentioned earlier, SPAS was equipped with a GPS receiver, and cross-linked a state vector identified as having a GPS origin. In an effort to somehow address GPS accuracy, a sawtooth comparison of the SPAS GPS state vector and the

orbiter INS state vector was performed during the period immediately after rendezvous[6]. Figures 8.4 and 8.5 show the sawtooth plots for these comparisons.

It is important to note that each data point on Figures 8.4 and 8.5 does not represent the receipt of two state vectors, but rather only one, either that from SPAS or from the orbiter. The NPS software was written such that when any new state vector arrives, the state vector that it is to be compared with is propagated with a high fidelity Cowell routine so as to match the time stamps.

Initially, the appearance of the sawtooth form in the position difference plot (Figure 8.4) is encouraging. However, a comparison with the velocity difference plot (Figure 8.5), and an examination of the times involved indicates a behavior that was not predicted. The sawteeth of Figure 8.4 occur on the order of minutes, much faster than any possible series of orbiter state vector updates. Furthermore, the rises in the sawteeth of Figure 8.4 directly correspond to periods of relatively poor velocity correlation. Further investigation reveals that the periods of increasing position error directly correspond to periods when no new GPS state vectors were being received. The points representing GPS state vector updates are actually the low points of each sawtooth in Figure 8.4. The increasing position errors are the result of attempting to

---

[6]The SPAS INS state vector was not used because it was no longer maintained after rendezvous.

move the GPS state vector, characterized by relatively large velocity error, ahead in time.

A pure GPS state vector is derived deterministically. That is, at a given instant, the receiver uses the information available from the satellites in view to produce the state vector. At another instant, it is using totally different information. There is no memory, or put another way, the state vector at time t_i has absolutely no bearing on the state vector at time t_{i+1}. Because velocity is a derivative, deterministic velocities carry one to two orders of magnitude more relative error than deterministic positions for non-military receivers [Ref. 2].

Figures 8.4 and 8.5 highlight the danger of using GPS derived state vectors directly from a non-military receiver. Non-military receivers purposely produce a certain amount of error so as to deny the use of GPS information for targeting. This is accomplished by creating timing errors in the signal being sent by the GPS satellites. While these errors in the position vector are not prohibitively large, velocity errors make propagation of the state vector exceedingly dangerous. Figure 8.4 shows that position error increases of 1000 feet can be achieved in a matter of minutes by propagating a GPS state vector.

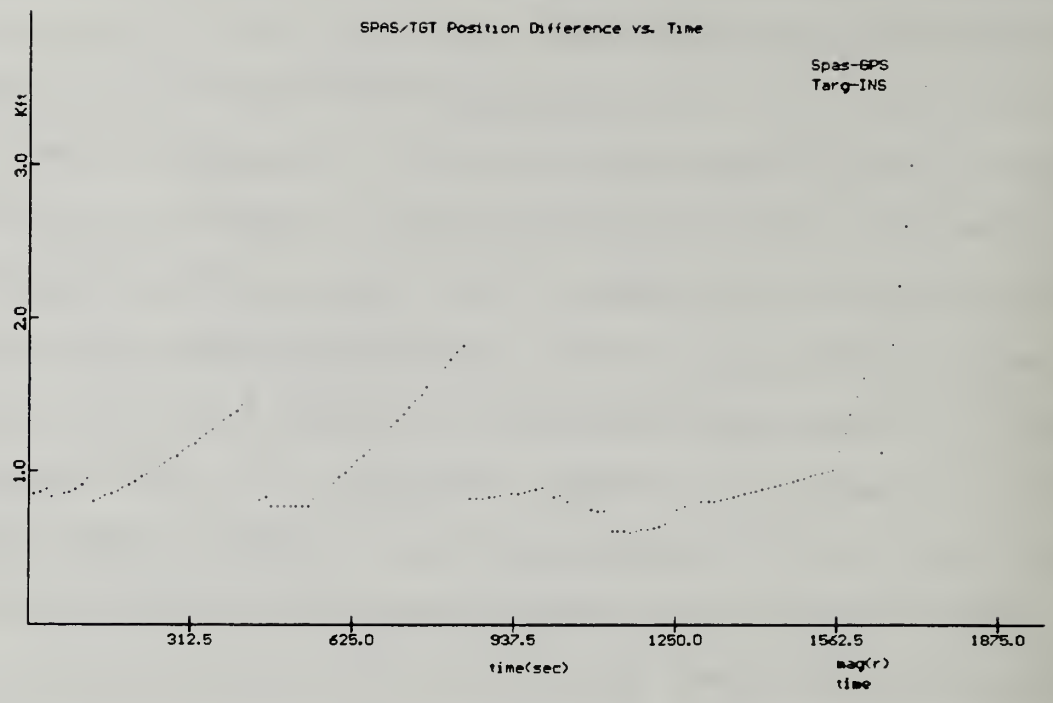

Figure 8.4 Orb-INS/SPAS-GPS Position Sawtooth Plot

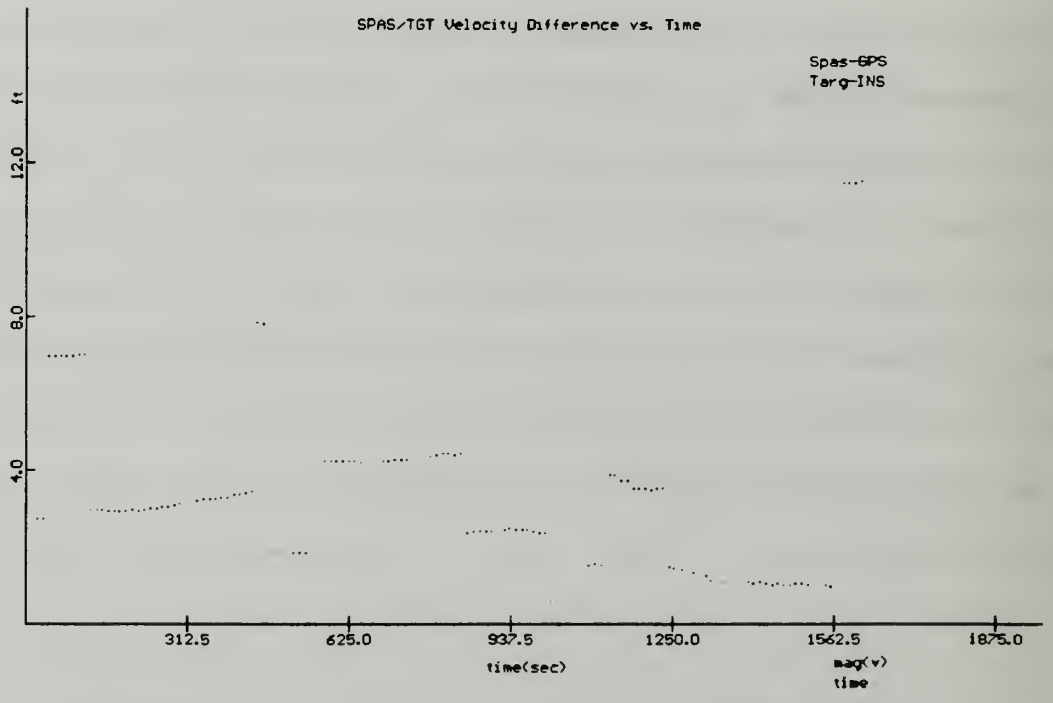

Figure 8.5 Orb-INS/SPAS-GPS Velocity Sawtooth

Clearly, for non-military GPS receivers, some form of filtering is necessary if state vectors are to be used for propagation. The NPS software has been modified to preclude the possibility of propagating a GPS state vector, and will remain so until a fast, recursive filter becomes available. A more thorough analysis of SPAS and TANS GPS information with filtering recommendations is the proposed thesis of Lieutenant Carolyn Tyler and Lieutenant Steve Rehwald. This thesis is due for publication in June 1994.

D. TDRSS VISIBILITY

The TDRSS constellation consists of two satellites in geosynchronous orbit. It's purpose is to provide a link between Low Earth Orbiting (LEO) spacecraft and ground controllers in the United States. The Shuttle, being a LEO spacecraft, communicates with Mission Control at the Johnson Space Center via a TDRSS link.

Ideally, when designing a geosynchronous constellation for global coverage, you would place three satellites in an equilateral triangle at the equator. The TDRSS constellation has this design, with the point of the missing satellite being over the Indian Ocean. This corresponds to a period of five to ten minutes of lost communications with Houston when the orbiter flies over the Indian Ocean. Figure 8.6 shows a portion of the downlinked data with gaps in the flight path.

These gaps correspond to the periods of lost communication over the Indian Ocean.

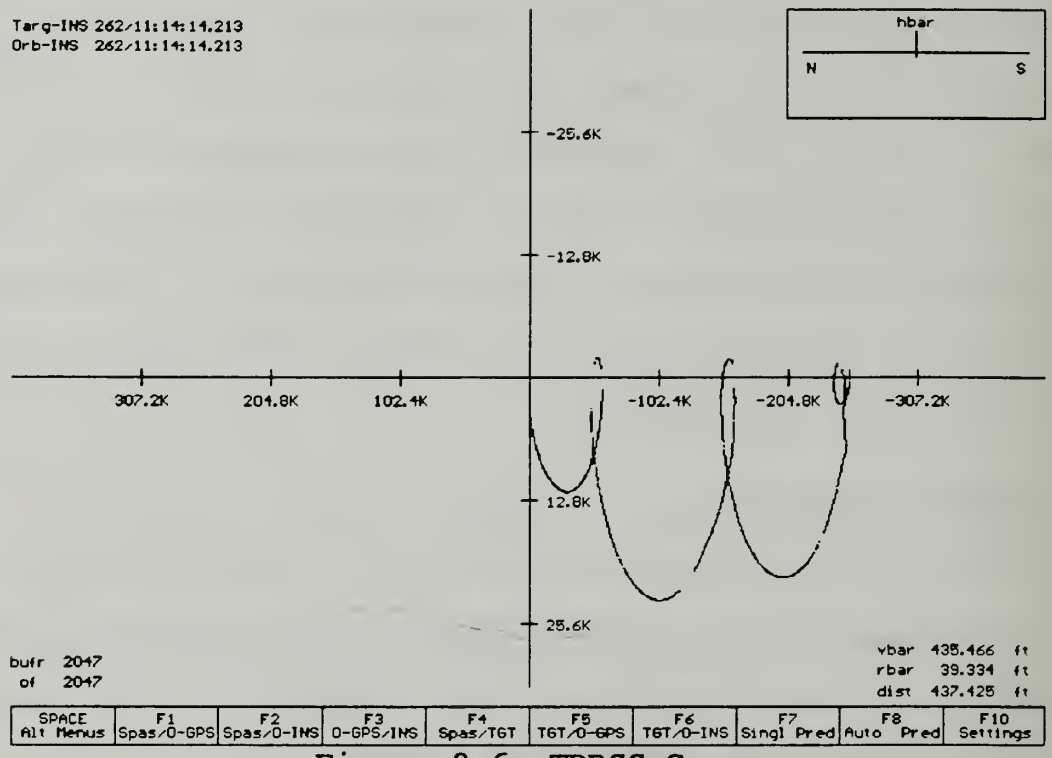

Figure 8.6 TDRSS Gaps

This phenomenon of periodic lost communication strengthens the argument for the continued use of tools such as the NPS software. Should an immediate decision need to be made during one of these lost communication periods, the flight crew would need access to the same types of information available to ground controllers.

IX. CONCLUSIONS

The theory presented in the previous chapters is fairly straightforward, and can be found in most Orbital Mechanics or Orbital Dynamics texts. The algorithms contained in the NPS software heavily exploit the works of the great mathematicians on the subject of the two body problem. Since the greats have passed, many others have sought improvements in orbit prediction through accounting for perturbing accelerations. The numerical algorithms required to attain these more "accurate" solutions were simply too computationally intensive for inclusion in the NPS software. The justification of the dismissal of perturbing accelerations is the one truly powerful statement of this thesis. It provided the opportunity for graduate students at the Naval Postgraduate School to have a *significant* impact on at least one manned space flight, and quite possibly, all NASA rendezvous operations in the future.

A. FLIGHT CREW REMARKS

Ideally, direct quotation from the *STS-51 FLIGHT CREW REPORT* would be appropriate. Unfortunately, the document is not available at time of writing. However, some excerpts from the crew inputs for the document, as well as many telephone conversations have provided some initial feedback on the applicability of the NPS software.

For the flight of STS-51, the certified rendezvous software was a program called "Payload Bay" (PLBAY). Unlike the NPS software, PLBAY is not automated, requiring manual inputs for most parameters. An automated version of PLBAY, called "Rendezvous Prox/Ops Program" (RPOP), also flew on STS-51. RPOP did not offer any new functionality over PLBAY, however it did greatly reduce the need for user interface. Some of the crews comments will use RPOP and PLBAY as a point of reference.

The sponsor of DTO 700-6/7 was Mission Specialist Dr. James Newman. The NPS software was his responsibility on orbit, thus he had the most favorable position for evaluating it's applicability. Dr. Newman has indicated that the NPS team "did an outstanding job, and should be really proud" of the NPS software. Furthermore he indicated that the NPS software was equally useful as a preflight training aid, and as a postflight debriefing tool [Ref. 8].

As stated in earlier chapters, the linearized equations of relative motion, and the Gauss problem, are both time constrained rendezvous solutions. The five planned burns prior to the visual takeover by the Mission Commander, are all time constrained rendezvous initiations. When asked if the NPS software should make these algorithms more accessible, Dr. Newman stated that this would make an ideal training tool, however, the question of validation as a *control* algorithm

would have to be addressed before consideration for flight certification [Ref. 8].

Dr. Newman was able to provide some of his inputs for the *STS-51 FLIGHT CREW REPORT* via facsimile [Ref 9]. Addressing the Future Thrust algorithm, he writes;

> ...These programs (PLBAY, RPOP, and NPS) ran from well prior to TI (Terminal Initiation, the first time constrained rendezvous) through the entire rndz and prox ops. The NPS code was able to perform future burns as well as "what if" thruster firings and *predicted* that the TI burn would result in a short rndz case. *This was born out* and MC1 though 4 all worked to correct this.

Comparing the NPS software to PLBAY and RPOP, he states;

> ...RPOP and the NPS plots were evaluated against the certified version of PLBAY and contributed to overall situational awareness. The NPS code had a number of features desirable in operational versions of RPOP, including the ability to select predictors more than 9 minutes in the future. It was also able to maintain the no-thrust predicted trajectory and the "what-if" trajectory at the same time, making comparisons of desired thrust inputs easier to do. And NPS kept track of the number of "what-if" firings in the various directions and the net delta-v in the orbiter axes.

Dr. Newman is also keen to point out that any algorithm is only as good as it's inputs;

> These tools, PLBAY, RPOP, and NPS, are only as good as the sensor data they receive, either directly as raw Ku radar data or indirectly in the filtered orbiter state vector. It is important to assess the quality of the sensor data, in this case the Ku radar data, before using the outputs of any of these programs.

Finally Dr. Newman recommends that NASA

> Incorporate desirable features from the NPS code into RPOP to improve situational awareness during the rendezvous and proximity operations.

The Mission Commander for STS-51, Captain Frank Culbertson was equally impressed with the performance of the NPS software. He too believes the software will be valuable as a training aid, and is looking forward to viewing the recorded flight data for the rendezvous with SPAS with the NPS code [Ref. 10].

Summarizing the impact of the NPS software, Captain Culbertson stated [Ref. 10];

> The product was outstanding, and gave insight not previously available. It was one more tool to help maintain the "big picture", and anything that increases situational awareness is valuable. A program (such as the NPS software) that processes data "real time" significantly enhances the crew's ability.

B. FOLLOW ON RESEARCH

Based on the commentary of the crew of STS-51, NASA is quite serious about the merger of the NPS software and RPOP. If any of the theory of the NPS software is to be validated by NASA, a detailed comparison of the theoretical differences between the NPS software and RPOP will be required.

1. Lambert Targeting

Although neglecting the effect of perturbing accelerations on a rendezvous solution produces very little error in the prediction of *relative* motion, error is still generated. Lambert targeting is designed to compensate for this error. The method requires iterations of a numerical high fidelity propagator and is therefor extremely computationally

intensive. The solution to the Gauss problem addressed in Chapter VII began with using the f and g functions to determine the target's position vector at the desired time of rendezvous, while Lambert targeting uses the high fidelity propagator to get this vector. Both methods then use the two body time constrained rendezvous solution to obtain the desired thrust. However, since the Lambert routine does not account for perturbing accelerations in the rendezvous solution, the calculated thrust will be somewhat in error. The Lambert routine then propagates (high fidelity) the chaser ahead with the input velocity changes and measures the error at the rendezvous end of the problem. The inverse of this error vector is then used as an offset aim point for another two body solution, and the chaser is again propagated ahead. The process is iterated until the "miss distance" is acceptably small [Ref. 7].

Since Lambert targeting is quite time consuming, the algorithm must be initiated well prior to the planned thrust. As stated in previous chapters, the first time constrained rendezvous solution occurs at TI (the last apogee before passing the target, see Figure 8.2). Thus for the period prior to TI, the passage of time corresponds to a decrease in distance to the target. Decreasing the distance to the target continually improves the relative error between the target and chaser state vectors via the KU band radar. It was this passage of time, and corresponding closure of the target, that

allowed for the NPS software to produce a better solution at TI than the Lambert targeting solution that was produced with inputs from a half an orbit prior to TI.

To completely address which method is best for computing the TI burn, a detailed analysis of the errors produced by the pure two body solution (like the methods of Chapter VII), and the errors produced by older inputs for Lambert targeting, is required. In performing this analysis, a further consideration is the ability of the crew to execute an exact rendezvous maneuver. Because thruster inputs have a finite number of digits available (typically down to tenths of a foot per second), the extra computation performed by the Lambert routine may well refine the solution beyond the point of input.

2. Inclusion of a Lambert Algorithm

Recognizing the reluctance to abandon an algorithm that has worked for many years, the NPS software could certainly be modified to include the Lambert algorithm. This inclusion could also serve to help in comparison of the two methods. At the very least, a significant speed up of the Lambert algorithm may be realized by using the output of the pure two body problem as a starting value.

3. Time Constrained Rendezvous Accessibility

The NPS software was not written with any particular standard maneuvers in mind. Specifically, when a time

constrained rendezvous solution is desired, the user must input when it is to start and stop. However the beginning and ending of all five of the time constrained algorithms can be calculated. The time at TI is that of the last apogee prior to passing the target, which is available via the f and g functions. The rendezvous is to be completed within 320°, which can be translated into a time via Kepler's equation, giving the start and stop time for the TI burn. The MC burns occur at fixed times relative to the TI burn, and can be assumed to complete a rendezvous at the same time as the TI burn. The NPS software should be modified to produce the time and thrusts required for the TI burn, as well as the MC burns.

4. Drag Accelerations

The accelerations caused by aerodynamic drag are a function of velocity and the size and shape of a vehicle. The argument for excluding drag accelerations when using the f and g functions, stems more from the insignificance of drag effects at the altitude of STS-51, than from the similarity of velocity vectors. If proximity operations are planned for a very low orbit, then drag accelerations must somehow be accounted for. With a constant atmosphere (Standard Day for instance) assumption, it is possible to estimate the altitude loss per unit time as a function of altitude, thus providing a fast analytic method for dealing with drag accelerations. The Future Plot and Future Thrust algorithms should be

flexible enough to operate in a very low altitude, high drag orbit.

5. Vernier Effect

Vernier effect is a term used to describe the apparent increase in energy of the shuttle's orbit over time. The cause is believed to be residual Δv from attitude control jets that are not perfect couples. After passing through the VBAR, just prior to rendezvous, the Future Plot algorithm showed that the orbiter would again reach the VBAR 110 feet in front of the target (Figure 9.1). The orbiter did not actually reach the

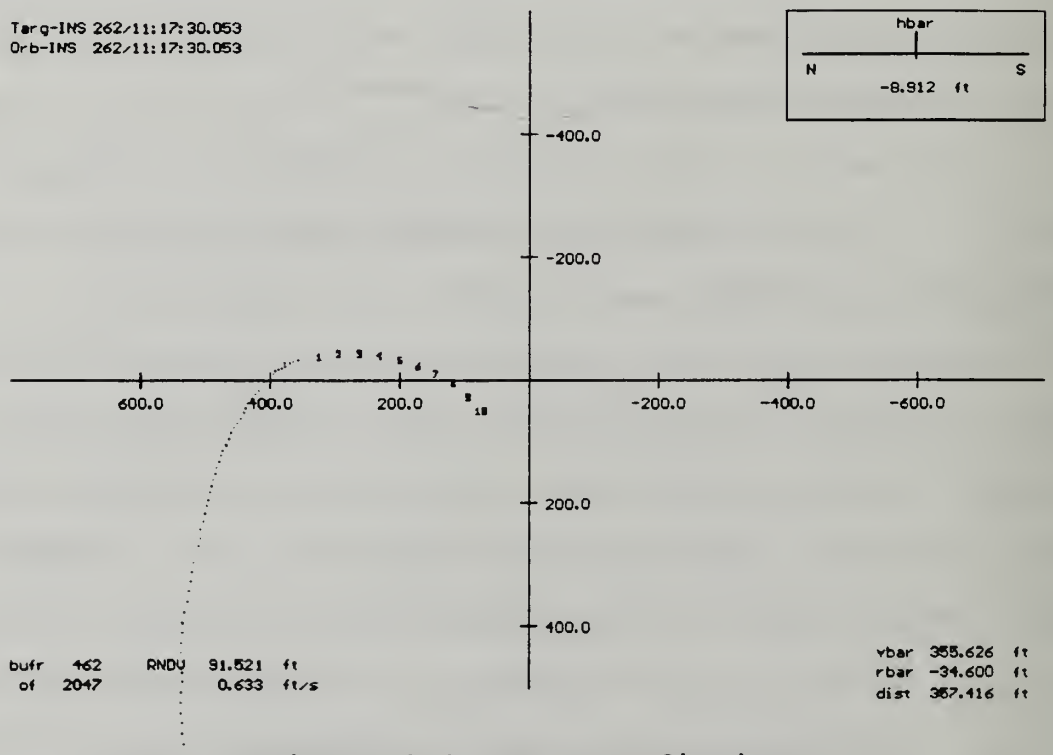

Figure 9.1 VBAR Prediction

VBAR until about 30 feet in front of the target [Ref. 10]. This 90 foot difference is quite significant when considering

the range to the target. During this period, the orbiter was maintaining a constant LVLH attitude, that is to say it was pitching with respect to inertial space. The attitude control thrusts required to hold this attitude are believed to be the cause of the excess energy. Currently, none of the software packages have a means of addressing this problem. If a means of quantifying the residual Δv becomes available, it should be incorporated into the Future Plot algorithm.

6. Target Attitude

Although the NPS software never addressed the target's attitude, target attitude information was available. During the rendezvous, Dr. Newman was observed using his right hand in a three axis orientation, trying to discern the target's attitude from pitch/yaw/roll angles provided by another source [Ref. 11]. Since the target quaternion is available, a graphical representation of target attitude with respect to LVLH or orbiter fixed space is possible. Captain Culbertson believes these pictures would be most helpful, as the mental gymnastics involved with deciphering the pitch/yaw/roll angles would no longer be necessary [Ref. 10].

C. SUMMARY

While the NPS software performed well, there is indeed room for improvement. Since the algorithms are not yet certified, access to the software for the purposes of making improvements is not difficult. Once the algorithms are

incorporated into a certified piece of software, they will become somewhat less accessible, and any changes will have to go through the validation process. Several changes have been made since the flight of STS-51, such as the inclusion of closure rate, out-of-plane rate, and improved graphics.

Participation on the NPS team necessitates direct contact with the next crew planning to use the NPS software, responding to their needs, and making improvements that will enhance the value of our product in the eyes of the user. It also provides an ideal opportunity for an experience tour in the Astronaut Office, gaining first hand exposure to the needs of a crew on orbit.

If the recommendations of the STS-51 crew are followed, some portions of the NPS software will live on in another program, and have an influence on manned space flight for years to come. However, the amount of influence rests on the continued relationship of the Naval Postgraduate School with the Astronaut Office at Johnson Space Center. It is my sincere hope that other students will follow our path, and continue to improve the NPS software, making it the invaluable all encompassing rendezvous/prox ops software that NASA seeks.

REFERENCES

1. Chobotov, V. A., *Orbital Mechanics*, American Institute of Aeronautics and Astronautics, 1991.

2. Saunders, P. et al., *The First Flight Test of GPS on the Space Shuttle*, Institute of Navigation, National Technical Meeting, 1994.

3. *Quaternions Supplemental Workbook*, QUAT-S 2102, Advanced Training Series, NASA, Johnson Space Center, 1984.

4. Danby, J. M. A., *Fundamentals of Celestial Mechanics*, Willman-Bell Inc., 1989.

5. Zeleny, W. B., 'Orbital Mechanics', Lecture notes for Introduction to Orbital Mechanics (PH2511), Naval Postgraduate School, 1990.

6. Bate, R. R., Mueller, D. D., and White, J. E., *Fundamentals of Astrodynamics*, Dover Publications Inc., 1971.

7. *Rendezvous/Proximity Operations Workbook*, RNDZ 2102, NASA Johnson Space Center, 1985.

8. Telephone conversation between Dr. James Newman, Astronaut Office (Code CB), NASA, Johnson Space Center, and the author, 21 November 1993.

9. Facsimile of inputs by Dr. James Newman for *STS-51 Flight Crew Report* (in work), Astronaut Office (Code CB), NASA, Johnson Space Center, 19 November 1993.

10. Telephone conversation between Captain Frank Culbertson, Astronaut Office (Code CB), NASA, Johnson Space Center, and the author, 21 November 1993.

11. Video Tape of STS-51 crew during the rendezvous of Space Shuttle Discovery with ORFEUS/SPAS, Astronaut Office (Code CB), NASA, Johnson Space Center, 19 September 1993.

INITIAL DISTRIBUTION LIST

		No. Copies
1.	Defense Technical Information Center Cameron Station Alexandria VA 22304-6145	2
2.	Library, Code 052 Naval Postgraduate School Monterey CA 93943-5002	2
3.	Clyde Scandrett (Code MA/Sd) Department of Mathematics Naval Postgraduate School Monterey CA 93943-5216	1
4.	Chairman, (Code MA) Department of Mathematics Naval Postgraduate School Monterey CA 93943-5216	1
5.	Chairman, (Code AA) Department of Aeronautics and Astronautics Naval Postgraduate School Monterey CA 93943-5000	1
6.	Chairman, (Code SP) Space Systems Academic Group Naval Postgraduate School Monterey CA 93943-5000	1
7.	Donald Danielson (Code MA/Dd) Department of Mathematics Naval Postgraduate School Monterey CA 93943-5216	1
8.	Beny Neta (Code MA/Nd) Department of Mathematics Naval Postgraduate School Monterey CA 93943-5216	1
9.	Tom Silva The Telemtry Workshop Inc. 17629 El Camino, Suite 206 Houston TX 77058	1

10. James Newman 1
NASA, Johnson Space Center
Astronaut Office - CB
Houston TX 77058

11. Brent Jett 1
NASA, Johnson Space Center
Astronaut Office - CB
Houston TX 77058

12. Dan Bursch 1
NASA, Johnson Space Center
Astronaut Office - CB
Houston TX 77058

13. Bill Readdy 1
NASA, Johnson Space Center
Astronaut Office - CB
Houston TX 77058

14. Frank Culbertson 1
NASA, Johnson Space Center
Astronaut Office - CB
Houston TX 77058

15. Jack Brassel 1
McDonnell Douglas Space Systems
Houston Division
13100 Space Center Blvd
Houston TX 77059-3556

16. Lee Barker (Code 31) 1
Space Systems Engineering
Naval Postgraduate School
Monterey CA 93943-5000

17. Brij Agrawal (Code AA/Ag) 1
Department of Aeronautics and Astronautics
Naval Postgraduate School
Monterey CA 93943-5000

DUDLEY KNOX LIBRARY
NAVAL POSTGRADUATE SCHOOL
MONTEREY CA 93943-5101

DUDLEY KNOX LIBRARY
3 2768 00307951 8